

***Étude d'algorithmes de suivi temporel
de segmentation basée mouvement
pour la compression de séquences d'images***

LAURENT BONNAUD, CLAUDE LABIT

N° 2253

janvier 1994

PROGRAMME 4



***apport
de recherche***

Étude d'algorithmes de suivi temporel de segmentation basée mouvement pour la compression de séquences d'images

LAURENT BONNAUD*, CLAUDE LABIT**

Programme 4 — Robotique, image et vision
Projet Temis

Rapport de recherche n° 2253 — janvier 1994 — 43 pages

Résumé : L'objet de cette étude est la compression de séquences d'images par compensation de mouvement. Afin d'utiliser les redondances temporelles inter-images, le mouvement entre l'image à coder et l'image précédente est estimé. Au moment de la décompression, les pixels de cette image sont reconstruits en fonction de leur intensité à la position qu'ils occupaient dans l'image précédente.

Dans cette étude, les images sont segmentées en régions homogènes au sens du mouvement et à frontière quelconque (polygone, B-spline). Dans chaque région, le mouvement est décrit par un modèle affine simplifié à 4 paramètres. Les techniques d'estimation des paramètres du mouvement nécessitant une bonne initialisation, un suivi long-terme par filtrage de Kalman des régions et des paramètres de leur mouvement semble donc approprié.

La mise en œuvre de cette méthode de codage comporte 3 étapes : prédiction par un filtre de Kalman des paramètres de mouvement de chaque région et de sa frontière, ajustement des frontières prédites par contours actifs (*snakes*) et estimation des paramètres de mouvement de chaque région. Des résultats sur séquences illustrent et valident le concept de suivi temporel de segmentation spatiotemporelle pour la compensation de séquences d'images.

(Abstract: *pto*)

*. IRISA e-mail bonnaud@irisa.fr

** IRISA e-mail labit@irisa.fr

Temporal linking of motion-based segmentation for image sequence coding

Abstract: This report describes an image sequence coding scheme with motion compensation. In order to use the inter-images redundancies in an image sequence, the motion between the image being coded and the previous reconstructed image is estimated. The motion compensated prediction errors are then quantized and transmitted. For image reconstruction, the dual algorithmic stages are performed.

In this study, images are segmented into homogeneous regions. The homogeneity criterion is based on a motion model (polynomial model) and the regions have an arbitrary shape (polygone or B-spline). In each region, the motion is described by a 4 parameters simplified affine model. As the estimation of the motion parameters requires a good initialisation and so locally can be biased, a long-term filtering of the motion parameters (based on temporal recursive Kalman formalism in our case) seems appropriate.

The implementation of this coding scheme consists in 3 steps : predicting the motion parameters and the shape of the regions with the Kalman filter, fitting the predicted spatio-temporal segmentation to the actual edges in the image and estimating the motion parameters of each region after spatiotemporal adjustment. This report proposes several results applied to temporal sequences to validate the introduction of such a temporal filtering.

Table des matières

1	Introduction	4
2	Cadre général	6
2.1	Compression de séquences d'images par compensation de mouvement	6
2.1.1	Champ dense des vecteurs vitesse/déplacement apparents 2D	6
2.1.2	Segmentation spatio-temporelle – Régions homogènes au sens du mouvement	7
2.2	Modèles de mouvement apparent 2D	8
2.2.1	Définition de quelques modèles de mouvement	8
2.2.2	Hierarchie de modèles de mouvement	9
2.3	Segmentation et estimation de mouvement	11
3	Présentation de la méthode de suivi	12
3.1	Aperçu de la méthode de suivi	12
3.1.1	Les 3 étapes principales de la méthode	12
3.1.2	Enchaînement des étapes du suivi long terme pour <i>une</i> région	12
3.2	Filtrage de Kalman des paramètres du mouvement	14
3.2.1	Formalisme général	14
3.2.2	Application au filtrage des paramètres de mouvement	15
3.3	Ajustement de la segmentation spatio-temporelle par contours actifs	16
3.3.1	Formulation mathématique du problème	17
3.3.2	Minimisation de l'énergie du <i>snake</i>	18
3.4	Estimation des paramètres du mouvement	18
3.5	Résolution des ambiguïtés	19
4	Résultats expérimentaux et extensions de l'algorithme	20
4.1	Présentation des séquences de test	20
4.2	<i>Snakes</i>	22
4.2.1	Implémentation	22
4.2.2	Paramètre de lissage	23
4.2.3	Exemples d'ajustement	24
4.2.4	Conclusion	27
4.3	Filtrage de Kalman	28
4.3.1	Limites de la prédiction par filtre de Kalman	28
4.3.2	Courbes de filtrage	31
4.3.3	Comparaison de plusieurs politiques de "projection"	33
4.4	Résultats globaux	34
4.4.1	Meilleure EQM moyenne région par région	34
4.4.2	Segmentation finale	34
5	Conclusion	37
6	Annexe A : courbes des paramètres mesurés, estimés et prédits par filtrage de Kalman	38
7	Annexe B : comparaison des "projections" CTS, CTL et LT	39

1 Introduction

Cette étude s'inscrit dans le cadre d'une convention de recherche intitulée "*Segmentation spatio-temporelle pour le codage de séquences télévisuelles*", passée entre le CCETT d'une part, l'IRISA et l'IRESTE d'autre part. Elle fait suite à la thèse de Henri Nicolas [Nic92] sur le codage de séquences d'images, ainsi qu'aux travaux de Victor Garcia-Garduño [GL93].

Elle s'inspire des travaux de François Meyer [Mey93] dans le domaine du suivi de régions pour l'analyse de scènes et de ceux de Bénédicte Bascle [Bas92] dans le domaine des contours actifs. Une chaîne logicielle complète de suivi temporel de régions, destinée à la compression de séquences d'images a donc été conçue. Initialement, Meyer et Bascle avaient conjointement développé une telle méthodologie [BM93] mais celle-ci était mono-objet et plutôt destinée à des applications d'analyse de scènes. L'application du suivi à la compression impose des contraintes supplémentaires, notamment sur la nécessité d'inverser le temps (pour la compensation de mouvement) et sur le choix d'un estimateur de mouvement qui vise uniquement à minimiser l'erreur de compensation de mouvement.

L'exploitation des résultats du suivi pour la compression par compensation de mouvement peut se faire de deux manières :

- Mode asymétrique : la carte de segmentation et les descripteurs de mouvement compacts sont transmis au décodeur dont la tâche est alors très simple et très rapide. Il lui suffit d'utiliser ces informations pour reconstruire l'image courante. L'inconvénient est la taille de ces informations qui n'est pas forcément aussi compacte qu'on le souhaiterait.
- Mode symétrique : le décodeur fait le même suivi que le codeur, sur les images déjà décodées. Il n'y a donc pas d'informations de segmentation et de mouvement à transmettre mais le décodeur, avec un délai d'une image, fonctionne de manière causale et ne peut donc disposer que d'une prédiction grossière de l'image à décoder. Il faut alors transmettre des ajustements de segmentation et de mouvement calculés par le codeur, ou transmettre une image de différences entre la prédiction du décodeur et la vraie image.

Ces indications ne sont données que pour illustrer l'intérêt d'un suivi temporel de segmentation spatio-temporelle ; le schéma de codage lui-même, justifiant une étude à part entière, n'a donc pas été abordé ici.

Les objectifs de cette étude furent :

- de valider l'utilisation d'un filtre de Kalman pour prédire le mouvement et la segmentation spatio-temporelle,
- de fournir une méthode alternative d'ajustement de segmentation spatio-temporelle par des contours actifs, en comparaison d'une technique utilisant un champ de Markov [GL93].

La première partie de ce rapport sera consacrée à la présentation des méthodes de compression par compensation de mouvement, puis aux modèles de mouvement servant à décrire le déplacement d'un objet entre 2 images. Quelques méthodes

classiques de segmentation spatio-temporelle seront ensuite brièvement rappelées, de manière à apprécier les différences avec notre algorithme.

Une deuxième partie présentera la version de base de l'algorithme, avec ses 3 modules principaux : filtrage de Kalman, ajustement de la segmentation prédite par contours actifs et estimation du mouvement. Enfin, la résolution des ambiguïtés qui apparaissent après la phase d'ajustement sera envisagée.

Une troisième partie exposera des résultats expérimentaux obtenus sur des séquences synthétiques et les extensions de l'algorithme qu'ils nous ont suggérées.

La conclusion portera sur les améliorations qui restent encore à apporter à la méthode.

2 Cadre général

2.1 Compression de séquences d'images par compensation de mouvement

Dans cette partie, nous allons présenter ce qu'est la compression avec compensation de mouvement et insister sur les différences qui existent entre un mouvement décrit pixel par pixel et un mouvement d'une région entière de l'image.

2.1.1 Champ dense des vecteurs vitesse/déplacement apparents 2D

Le principe de la compensation de mouvement est de reconstruire les pixels de l'image courante à partir d'une prédiction basée sur la valeur d'intensité qu'ils avaient à une position déplacée dans l'image précédente. Pour cela, on définit le vecteur déplacement/vitesse apparent 2D pour chaque point¹ de l'image. Soit un point p qui se déplace entre les images I_t et I_{t+1} de la position p_t à la position p_{t+1} . Son vecteur déplacement/vitesse apparent 2D est (voir la figure 1) :

- dans le sens du temps croissant $\vec{d}_t^+(p) = p_{t+1} - p_t$. Ce sens sert pour la prédiction de la segmentation spatio-temporelle de I_t vers I_{t+1} et pour le filtrage de Kalman que l'on fait dans ce sens causal, même si un filtrage pouvait être envisagé dans l'autre sens.
- dans le sens du temps décroissant $\vec{d}_{t+1}^-(p) = p_t - p_{t+1}$. Ce sens sert pour la compensation de mouvement comme nous allons le voir immédiatement.

La version reconstruite de I_{t+1} étant désignée par \hat{I}_{t+1} , l'intensité d'un pixel reconstruit $\hat{I}_{t+1}(p) = \hat{I}_t(p + \vec{d}_{t+1}^-(p))$ est calculée par interpolation (bilinéaire ou autre) dans \hat{I}_t . Remarquons que le décodeur ne peut interpoler directement dans I_t car il ne dispose que de la version reconstruite. Un codeur complet se devrait d'en tenir compte, mais comme notre module de suivi est dépourvu de la phase ultérieure de codage, il ne peut pas utiliser \hat{I}_t et se contente donc de I_t . On fait l'hypothèse que cette approximation ne va pas entacher nos résultats d'un biais trop important. L'hypothèse nécessaire à la bonne reconstruction de I_{t+1} est que l'intensité lumineuse d'un point ne change pas d'une image sur l'autre. Cette hypothèse est vérifiée dans beaucoup de séquences d'images courantes, mais examinons quelques situations où ce n'est pas le cas :

- L'éclairage de la scène varie, l'ouverture de l'objectif change, ou il y a un fondu par le noir entre deux séquences, introduisant une variation d'illumination dans

1. On distingue :

- le pixel qui est un rectangle dans le plan image de la caméra. Celle-ci obtient le niveau de gris affecté au pixel à partir de l'intensité lumineuse sur ce rectangle.
- le point qui est un élément de l'image, considérée comme une partie rectangulaire de \mathbb{R}^2 .

Pour chaque pixel, on définit un point représentant qui peut être l'un de ses coins ou son centre. La valeur d'intensité d'un point s'obtient par interpolation entre les valeurs des pixels adjacents. Cette précision est importante pour la localisation, dans l'espace dual de celui défini par les pixels, des frontières des régions.

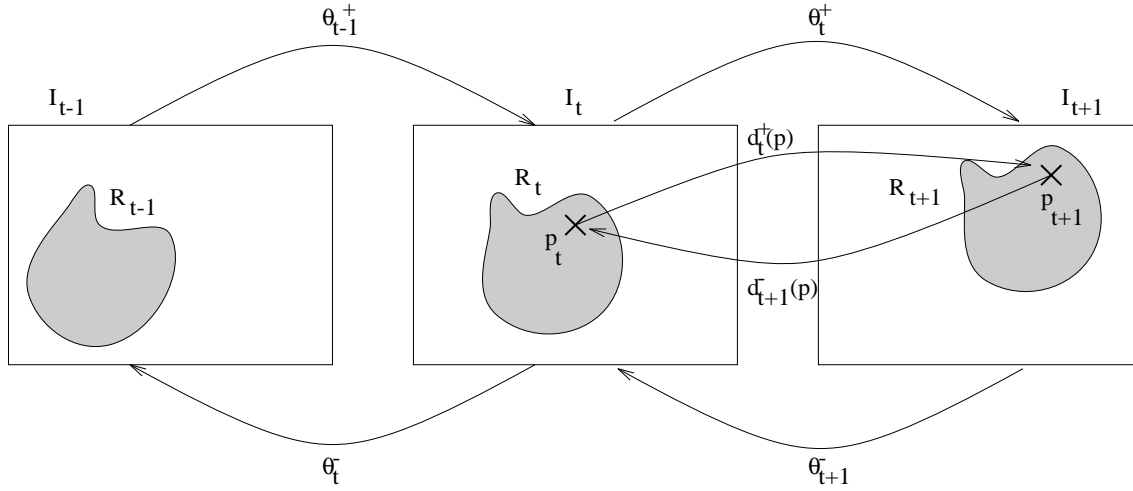


FIG. 1 - Les deux sens de description du mouvement.

l'image projetée. Ce problème a été envisagé par [Nic92] [NKL93] mais n'a pas été retenu dans cette étude.

- Il y a un raccord entre 2 séquences à contenu de scène différent.
- Des objets transparents colorés ou semi-transparentes s'interposent entre la caméra et les autres objets de la scène.
- Les paramètres intrinsèques de la caméra changent. Par exemple, si la mise au point varie, certains objets deviennent flous et d'autres nets.
- La couleur des objets change (le problème ne se pose pas ici, car nous n'étudions que des images en niveaux de gris).
-

Le champ dense des vecteurs $\vec{d}_{t+1}^-(p)$, représentant un volume de données trop important, il n'est pas transmis tel quel au décodeur pour chaque pixel de l'image. Il est préférable de segmenter la scène en régions homogènes au sens du mouvement et pour chaque région, d'identifier et de transmettre un descripteur de mouvement apparent 2D, noté $\Theta_{\mathcal{R},t+1}^-$ (ou sa différence avec un descripteur prédit par le décodeur dans le cas d'un mode de fonctionnement symétrique du codeur et du décodeur) utilisable pour tous les pixels de la région \mathcal{R}_{t+1} .

2.1.2 Segmentation spatio-temporelle – Régions homogènes au sens du mouvement

Une région \mathcal{R} est dite *homogène au sens du mouvement* quand le champ dense des déplacements apparents 2D de ses points $\vec{d}^\pm(p)$ dérive d'un descripteur de mouvement 2D unique $\Theta_{\mathcal{R}}^\pm$ (par exemple : rotation, translation ou homothétie de la région entière) :

$$\forall p \in \mathcal{R}, \vec{d}^\pm(p) = \mathcal{M}(\Theta_{\mathcal{R}}^\pm, p)$$

où \mathcal{M} désigne un certain modèle de mouvement (voir § 2.2).

L'intérêt pour la compensation de mouvement est que le déplacement de chaque pixel d'une région pourra être calculé à partir du descripteur de mouvement 2D compact $\Theta_{\mathcal{R},t+1}^-$. Seul $\Theta_{\mathcal{R},t+1}^-$ est transmis (ou sa différence avec un descripteur prédit), typiquement un vecteur d'une dizaine de composantes, au lieu de $\vec{d}_{t+1}^-(p)$. Le gain apparaît donc nettement dès que la taille de la région est supérieure à quelques dizaines de pixels.

Par contre, il faut aussi segmenter I_{t+1} en régions homogènes au sens du mouvement et transmettre cette segmentation (ou son écart avec une segmentation prédite). Certains schémas de compression contournent la difficulté en choisissant une segmentation convenue à l'avance entre le codeur et le décodeur. Par exemple, dans MPEG [Gal91][Gal92], il s'agit de carrés de taille fixe. Dans cette étude, nous nous intéressons à des segmentations moins frustes, où chaque région est délimitée par une courbe polygonale ou par une B-spline. Ce type de segmentation a l'avantage de pouvoir se suivre dans le temps, ce qui n'est pas le cas avec des blocs (MPEG) ou même avec un *quadtree* [Nic92].

2.2 Modèles de mouvement apparent 2D

Cette partie est consacrée à la notion de modèle de mouvement et à la présentation d'une hiérarchie de modèles de mouvement.

2.2.1 Définition de quelques modèles de mouvement

Il s'agit de modèles qui associent un champ dense de vecteurs vitesse apparente 2D $\vec{d}_{\mathcal{R}}^\pm(p)$ à un descripteur de mouvement $\Theta_{\mathcal{R}}^\pm$ d'une région. Ils résultent :

- d'une approximation plus ou moins grossière du mouvement d'un objet de l'espace, projeté dans le plan de la caméra, ou
- du choix *a priori* d'un modèle polynomial à 2 indéterminées (les coordonnées des pixels) [LP80][Adi85].

Il y a parfois coïncidence entre ces deux approches qui fournissent des modèles équivalents (modèle affine). Une fois le modèle choisi, le descripteur de mouvement permet de calculer le vecteur déplacement de chaque pixel de la région grâce à la formule générale suivante :

$$\vec{d}_{\mathcal{R}}^\pm \begin{pmatrix} x \\ y \end{pmatrix} = \mathcal{M} \left[\Theta_{\mathcal{R}}^\pm, \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix} \right]$$

Le mouvement est systématiquement repéré par rapport à un point G qui est le centre de gravité de la région suivie, tant qu'il n'y a pas d'occlusion. En cas d'occlusion, le nouveau centre de gravité n'est pas recalculé, mais G est astreint à se déplacer avec la région (voir § 3.5). Les raisons de ce choix seront expliquées lors de la description du modèle utilisé (voir § 2.2.2).

Dans la suite, les notations $\Theta_{\mathcal{R}}^\pm$ et $\vec{d}_{\mathcal{R}}^\pm$ sont allégées en Θ et \vec{d} .

2.2.2 Hiérarchie de modèles de mouvement

Il s'agit de modèles de plus en plus riches, avec de plus en plus de degrés de liberté. Cette hiérarchie peut être parcourue dans le sens ascendant ou descendant car les coefficients d'un modèle simple se retrouvent identiquement dans le modèle immédiatement supérieur. Le modèle le plus simple qui fournit une description satisfaisante du mouvement peut être choisi selon les critères statistiques de Rissanen [Ric92].

Modèle nul (0 paramètres) $\Theta = \vec{0}$

$$\vec{d}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Une région suit ce modèle quand elle n'est pas en mouvement (fond immobile par exemple). Utiliser ce modèle revient à détecter les zones de I_{t+1} qui sont identiques dans I_t et à faire un rafraîchissement conditionnel. Ce modèle est utilisé quand le temps de compression/décompression doit être très faible. Par exemple, il est utilisé dans le système de compression *Cell* [Sun93], dont le principal objectif est le décodage en temps réel sur une station de travail courante, sans ajout d'accélérateur spécifique.

Modèle constant (2 paramètres) $\Theta = [t_x, t_y]$

$$\vec{d}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Ce modèle prend en compte les régions qui ont un mouvement de translation uniforme. C'est celui qui est utilisé dans la norme MPEG [Gal91][Gal92], en association avec une segmentation en blocs. Cependant, MPEG se limite à $(t_x, t_y) \in \mathbb{Z}^2$ ou à $(2t_x, 2t_y) \in \mathbb{Z}^2$, alors qu'ici, $(t_x, t_y) \in \mathbb{R}^2$ et la quantification est faite ultérieurement.

Modèle affine simplifié (4 paramètres) $\Theta = [t_x, t_y, k, \theta]$

$$\vec{d}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix} \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix}$$

Ce modèle prend en compte les régions qui ont un mouvement de translation, de rotation, ou qui subissent une homothétie. C'est celui qui a été utilisé dans cette étude, car il réalise un bon compromis entre la richesse de description du mouvement des modèles complexes et la sur-segmentation introduite par les modèles trop simples [IRE92].

Le paramètre θ est appelé *angle de "rotation"*, k est appelé *paramètre de "divergence"* et $\lambda = 1 + k$ est un *"rapport d'homothétie"*. Ceci est justifié par une approximation au 1^{er} ordre d'une matrice M représentant une composition de rotation et d'homothétie :

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} \lambda \cos \theta & -\lambda \sin \theta \\ \lambda \sin \theta & \lambda \cos \theta \end{bmatrix}$$

En effet, quand θ et k sont faibles (petit mouvement), les approximations suivantes sont justifiées :

$$\begin{cases} \cos \theta & \sim 1 \\ \sin \theta & \sim \theta \\ \lambda & \sim 1 \end{cases}$$

Ainsi, la matrice de notre modèle qui vaut $M - I_2$ s'approxime en $\begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix}$. Toutefois, cette approximation ne restreint pas la généralité du modèle : si l'on veut représenter exactement la rotation d'angle θ' et l'homothétie de rapport λ' , il suffit de prendre

$$\begin{cases} \lambda & = \lambda' \cos \theta' \\ \theta & = \lambda' \sin \theta' \end{cases}$$

[Nic92] a montré que choisir G confondu avec le centre de gravité de la région suivie, donne une meilleure estimation (voir § 3.4). Changer de point de référence ne change pas les paramètres de la matrice $\begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix}$ mais a une influence sur les paramètres de translation $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$. G est le point dont le vecteur de déplacement vaut $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$. Choisir G fixe et arbitraire (un coin ou le centre de l'image par exemple) donnerait des paramètres de translation qui ne correspondraient pas au déplacement perçu de la région. Il semble donc plus naturel de le choisir proche du centre de gravité pour que les résultats sur la translation soient facilement interprétables. Nous verrons surtout que ce détail a son importance pour le filtrage de Kalman des paramètres de translation (voir § 3.2). Nous verrons aussi comment choisir ce point G lorsque la région subit une occlusion (voir § 3.5).

Modèle affine (6 paramètres) $\Theta = [t_x, t_y, a, b, c, d]$

$$\vec{d} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix}$$

Ce modèle rajoute au précédent des déformations comme les affinités et les cisaillements. C'est celui qui a été utilisé par [Fra91][Mey92][Mey93] et [BM93].

Modèle homographique (8 paramètres)

$$\Theta = [N_{xx}, N_{xy}, N_{yx}, N_{yy}, N_x, N_y, D_x, D_y]$$

$$\vec{d} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{N_{xx}(x-x_G)+N_{xy}(y-y_G)+N_x}{D_x(x-x_G)+D_y(y-y_G)+1} \\ \frac{N_{yx}(x-x_G)+N_{yy}(y-y_G)+N_y}{D_x(x-x_G)+D_y(y-y_G)+1} \end{pmatrix}$$

Ce modèle est capable de décrire exactement tout mouvement d'un objet 3D plan projeté sur la rétine de la caméra. Il est donc intéressant pour des scènes comportant des objets artificiels plans (immeubles, routes...). Il a l'inconvénient de ne pas s'inscrire parfaitement dans la hiérarchie. Il n'a donc pas été envisagé par [Nic92].

Modèle quadratique (12 paramètres)

$$\Theta = [t_x, t_y, a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5]$$

$$\begin{aligned} \vec{d} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix} \\ &+ \begin{bmatrix} a_3 & a_4 \\ b_3 & b_4 \end{bmatrix} \begin{pmatrix} (x - x_G)^2 \\ (y - y_G)^2 \end{pmatrix} + \begin{bmatrix} a_5 \\ b_5 \end{bmatrix} (x - x_G)(y - y_G) \end{aligned}$$

Ce modèle est identique au modèle affine avec des termes supplémentaires du second degré. Il est rarement utilisé car les termes quadratiques sont difficiles à estimer.

2.3 Segmentation et estimation de mouvement

Ce problème est central dans presque tous les traitements de séquences d'images et de nombreuses méthodes existent pour le résoudre. Cette partie va en rappeler brièvement quelques unes et faire la transition avec la description de notre méthode de segmentation/estimation.

Rappelons qu'il y a dualité entre segmentation spatio-temporelle et estimation des paramètres du mouvement ; ces 2 problèmes sont en effet à résoudre simultanément :

- Pour pouvoir calculer les paramètres du mouvement, il faut savoir sur quelle région les estimer.
- Pour pouvoir déterminer la segmentation spatio-temporelle, il faut savoir quels pixels ont un mouvement homogène et donc, il faut connaître leur mouvement.

De nombreuses méthodes utilisent le même schéma de résolution. Elles partent d'une segmentation initiale plus ou moins arbitraire :

- pixels individuels (techniques Markoviennes fournissant un champ dense)
- petits blocs de pixels (approche MPEG)
- grille arbitraire (approche de Henri Nicolas [Nic92])
- image entière
- segmentation spatiale (notre approche)
-

Elles estiment les paramètres du mouvement sur cette segmentation, puis réalisent des fusions et des divisions de régions selon un critère d'uniformité de mouvement. Ce processus est itéré jusqu'à la convergence.

Par exemple, [Nic92] part d'une grille de segmentation arbitraire et la subdivise selon un *quadtree* jusqu'à ce que les carrés obtenus soient bien compensés.

Dans le cadre de notre étude, l'approche consiste à partir d'une segmentation spatiale (issue des travaux de l'IRESTE [IRE92]) qui est plus fine que la segmentation spatio-temporelle. Notre étude concerne uniquement le suivi temporel de ces régions, qui va maintenant être présenté. Les fusions éventuelles et l'apparition de nouvelles régions sont décrites dans [IRE93a][IRE93b].

3 Présentation de la méthode de suivi

3.1 Aperçu de la méthode de suivi

Dans cette partie, nous allons exposer la façon dont s'enchaînent les étapes de la méthode pour le suivi d'une région. Les détails internes à chaque étape seront présentés dans les parties suivantes (voir § 3.2, § 3.3 et § 3.4). Ensuite, nous verront comment résoudre les problèmes liés au suivi de plusieurs régions simultanément (voir § 3.5).

3.1.1 Les 3 étapes principales de la méthode

Pour compresser l'image I_{t+1} , on enchaîne les 3 étapes suivantes :

- **prédiction** par un filtre de Kalman des paramètres de mouvement Θ_t^+ de chaque région et de sa frontière dans I_{t+1}
- **ajustement** des frontières prédites par contours actifs (*snakes*)
- **estimation** des paramètres de mouvement Θ_{t+1}^- de chaque région pour la compensation de mouvement

3.1.2 Enchaînement des étapes du suivi long terme pour une région

Ces différents éléments s'articulent ainsi : pour coder une région \mathcal{R}_{t+1} de l'image I_{t+1} :

0. Si $t = 0$, la segmentation spatio-temporelle est initialisée sur une segmentation spatiale de l'image I_1 et le descripteur de mouvement à zéro.
1. Partant du contour spatial initial ou de la frontière prédite de \mathcal{R}_{t+1} , cette frontière est ajustée par transformation en *snake*. Elle se recalcule ainsi sur les zones de fort gradient spatial dans I_{t+1} (les contours spatiaux), mais sa déformation est contrainte à faire partie de la hiérarchie de modèles de mouvement (affine, affine simplifié...). On obtient ainsi une information de mouvement appelée $\Theta_{t,DCS}^+$ (Déformation Contrainte du Snake) et la région \mathcal{R}_{t+1} .
2. Le mouvement $\Theta_{t+1,EQM}^-$ (au sens de l'Erreur Quadratique Moyenne) de \mathcal{R}_{t+1} est mesuré par une technique de minimisation de l'EQM par descente de gradient (programme de Henri Nicolas). On peut choisir comme initialisation soit $\Theta_{t/t-1}^+$ l'estimateur *a priori* du filtre de Kalman à $t - 1$ (lire " Θ^+ de t sachant $t - 1$ ") soit $\Theta_{t,DCS}^+ \circ \Theta_{t/t-1}^+$ (composition du mouvement prédit et de son ajustement).
3. Cette mesure de mouvement est filtrée : on obtient l'estimateur *a posteriori* du filtre de Kalman $\Theta_{t/t}^+$ (lire " Θ^+ de t sachant t ").
4. On calcule $\Theta_{t+1/t}^+$ (nouvel estimateur *a priori* pour le couple d'images suivant).
5. Il est appliqué à la frontière de \mathcal{R}_{t+1} pour obtenir une prédiction de la frontière de \mathcal{R}_{t+2} ("projection") qui permet de reboucler sur le point 1.

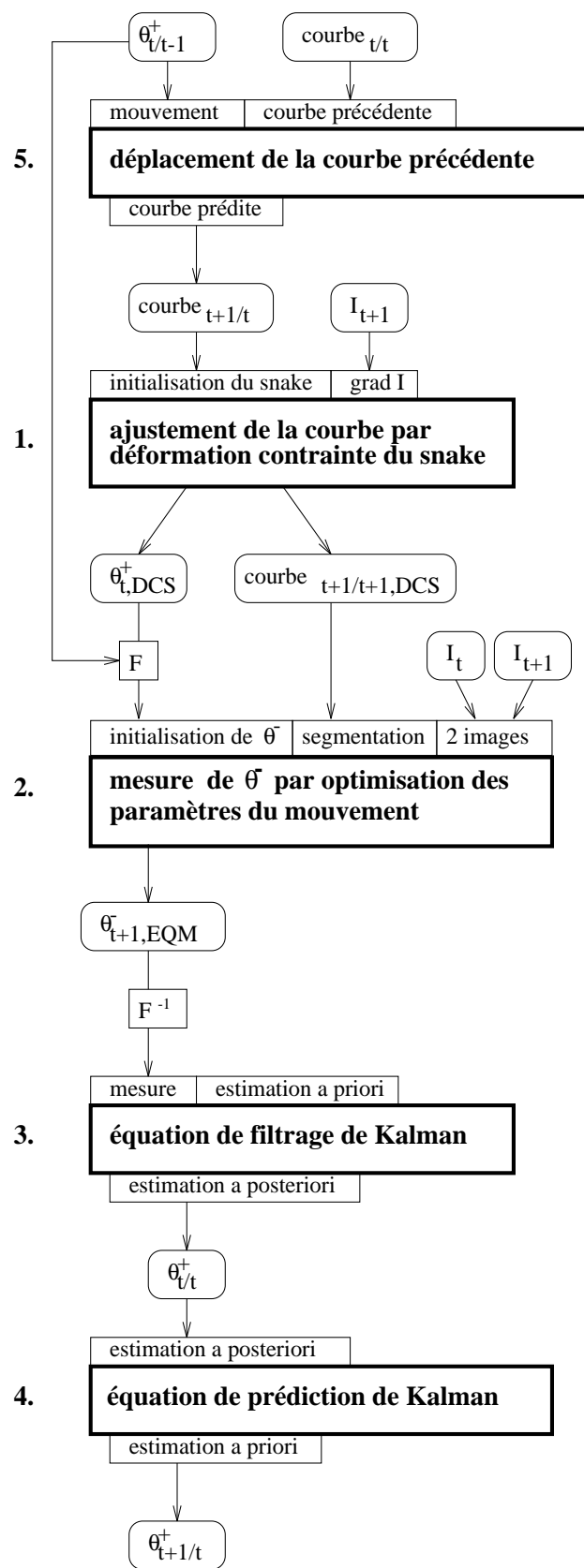


FIG. 2 - Schéma du suivi long terme pour une région.

3.2 Filtrage de Kalman des paramètres du mouvement

Les 2 idées qui motivent l'utilisation d'un filtre de Kalman sur les paramètres du mouvement sont :

- accumuler les mesures des paramètres de mouvement d'une région au cours du temps pour affiner leur estimation,
- obtenir une bonne prédiction du mouvement, nécessaire pour prédire la segmentation avant ajustement (voir § 3.3) et pour initialiser l'estimateur de mouvement (voir § 3.4).

Après avoir vu dans la partie précédente comment est utilisé le filtre de Kalman, rentrons en détail dans son fonctionnement.

3.2.1 Formalisme général

Le filtrage de Kalman consiste à estimer de façon optimale la valeur d'un vecteur d'état Ψ_t (espérance et matrice de covariance) sachant qu'à chaque instant on dispose d'un vecteur de mesure s_t qui est une combinaison linéaire bruitée des éléments de Ψ_t . L'algorithme utilise aussi la dynamique du vecteur d'état, bruitée de même. Il est optimal dans le sens où il prend en compte toutes ces informations et les combine au mieux de sorte à obtenir une estimation de variance minimale [Med69][Ang92][Gel74].

Équations d'évolution et d'observation

$$\begin{cases} \Psi_{t+1} &= A_t \Psi_t + w_t & (\text{dynamique bruitée du système}) \\ s_t &= H_t \Psi_t + v_t & (\text{mesures bruitées}) \end{cases}$$

avec w_t et v_t deux bruits blancs indépendants tels que :

- $E[v_t] = 0$
- $E[w_t] = 0$
- $E[v_{t_1} v_{t_2}^T] = R \delta_{t_1 t_2}$
- $E[w_{t_1} w_{t_2}^T] = Q \delta_{t_1 t_2}$
- $E[v_t w_t^T] = 0$

où Q et R sont 2 matrices symétriques définies positives. R caractérise le bruit de mesure et Q le bruit de modèle.

Équations de filtrage et de prédiction Connaissant la mesure s_t (et toutes les mesures antérieures), le problème est alors de trouver le meilleur estimateur

- de Ψ_t : il est noté $\Psi_{t/t}$ et s'appelle estimateur *a posteriori*. Notons $P_{t/t}$ sa matrice de covariance.

- de Ψ_{t+1} : il est noté $\Psi_{t+1/t}$ et s'appelle estimateur *a priori*. Notons $P_{t+1/t}$ sa matrice de covariance.

On définit par ailleurs K_t , appelé gain de Kalman qui pondère l'apport de l'innovation $s_t - H_t \Psi_{t/t-1}$ (différence entre l'observation et l'estimée *a priori* de cette observation) par rapport à la prédiction précédente $\Psi_{t/t-1}$.

Ces quantités sont données par les équations suivantes :

$$\begin{cases} K_t &= P_{t/t-1} H_t^T (R + H_t P_{t/t-1} H_t^T)^{-1} \\ \Psi_{t/t} &= \Psi_{t/t-1} + K_t (s_t - H_t \Psi_{t/t-1}) \\ P_{t/t} &= (I - K_t H_t) P_{t/t-1} \end{cases} \quad (\text{équations de filtrage})$$

$$\begin{cases} \Psi_{t+1/t} &= A_t \Psi_{t/t} \\ P_{t+1/t} &= A_t P_{t/t} A_t^T + Q \end{cases} \quad (\text{équations de prédiction})$$

Initialisation Il faut aussi fournir le vecteur d'état initial $\Psi_0 = \Psi_{0/-1}$ et sa matrice de covariance $P_0 = P_{0/-1}$.

3.2.2 Application au filtrage des paramètres de mouvement

Vecteurs d'état et d'observation Le modèle d'évolution du système est un mouvement à accélération constante. Si l'un quelconque des paramètres de mouvement $(t_x^+, t_y^+, \theta^+, k^+, a^+, b^+, c^+, d^+ \dots)$ à l'instant t est noté ψ_t , on ne pose pas $\ddot{\psi}_t = 0$, mais cette dérivée est modélisée par un bruit Gaussien centré en 0.

Le vecteur d'état est

$$\Psi_t = \begin{pmatrix} \psi_t \\ \dot{\psi}_t \\ \ddot{\psi}_t \end{pmatrix}.$$

Les différents paramètres de mouvement sont filtrés de façon indépendante : autant de filtres que $\Theta_{\mathcal{R}}^+$ a de composantes fonctionnent en parallèle. Une autre possibilité aurait été de regrouper tous les paramètres du modèle de mouvement et leurs dérivées dans un même vecteur d'état. Nous l'avons écartée pour des raisons de simplicité opératoire.

Le vecteur d'observation est

$$s_t = (\psi_t)$$

car ψ_t est la seule sortie du système que l'on peut mesurer (estimation du mouvement). Les variables $\dot{\psi}_t$ et $\ddot{\psi}_t$ sont considérées comme non observables.

Équations d'évolution et d'observation

- A_t est constante au cours du temps et vaut $A = \begin{bmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$
- H_t est constante au cours du temps et vaut $H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$
- w_t et v_t deux bruits Gaussiens de matrices de covariance Q et R .

Initialisation Le filtre ne démarre pas à $t = 0$ mais à partir de $t = 2$. [Mey92] a montré que

- $Q = \sigma_w^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}$, T étant la période d'acquisition des images.

Il suffit donc de fournir une seule variance σ_w pour chaque paramètre filtré.

- $\Psi_2 = \begin{pmatrix} \psi_2 \\ \frac{\frac{3}{2}\psi_2 - 2\psi_1 + \frac{1}{2}\psi_0}{T} \\ \frac{\psi_2 - 2\psi_1 + \psi_0}{T^2} \end{pmatrix}$

- $P_2 = \begin{bmatrix} \sigma_v^2 & \frac{3}{2T}\sigma_v^2 & \frac{1}{T^2}\sigma_v^2 \\ \frac{3}{2T}\sigma_v^2 & \frac{T^3}{3}\sigma_w^2 + \frac{13}{2T^2}\sigma_v^2 & \frac{9T^2}{40}\sigma_w^2 + \frac{6}{T^3}\sigma_v^2 \\ \frac{1}{T^2}\sigma_v^2 & \frac{9T^2}{40}\sigma_w^2 + \frac{6}{T^3}\sigma_v^2 & \frac{23T}{30}\sigma_w^2 + \frac{6}{T^4}\sigma_v^2 \end{bmatrix}$

Importance du choix du point G Nous avons vu que le choix du point G affectait la valeur des paramètres de translation. Par exemple, pour un objet en rotation de vitesse constante autour de son centre de gravité, repérer son mouvement par rapport à son centre de gravité donne des paramètres de translations nuls ; seul le paramètre de rotation est non nul. Par contre, si un autre point G était choisi distant de d , le vecteur $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$ décrirait un “cercle”² de rayon d . Le modèle d'évolution à accélération constante du filtre de Kalman ne prend pas bien en compte de telles variations.

3.3 Ajustement de la segmentation spatio-temporelle par contours actifs

Après la phase de prédiction, nous disposons donc d'un mouvement prédit que nous appliquons au *snake* qui entoure la région suivie. Cette segmentation prédite n'est en général pas parfaite. Il est donc nécessaire de l'ajuster aux contours spatio-temporels présents dans l'image traitée (I_{t+1}). Pour cela, 2 approches concurrentes ont été envisagées :

- Un ajustement par champ de Markov qui travaille au niveau des pixels et se fait pour toutes les régions simultanément. Il peut utiliser à la fois des informations spatiales dans l'image I_{t+1} et des informations de mouvement entre les images I_{t+1} et I_t [GL93].
- Un ajustement par contours actifs qui se fait sur *une* région à la fois et agit uniquement sur la frontière de celle-ci. Cette méthode n'utilise que des informations spatiales dans l'image I_{t+1} et éventuellement une régularisation de la forme du *snake*. Le *snake* évolue donc de sorte à se coller aux zones de forts gradient spatial (les contours). Dans notre étude, seul le gradient spatial a été utilisé, mais d'autres possibilités sont à envisager (par exemple des statistiques sur la luminance des pixels de la région le long de la courbe [Del93]).

2. Au sens de l'approximation d'un petit mouvement affine simplifié (voir § 2.2.2)

C'est cette deuxième approche que nous allons maintenant décrire.

L'hypothèse nécessaire au bon fonctionnement de cette méthode est que la segmentation spatio-temporelle soit un sous-ensemble de la segmentation spatiale. Cette hypothèse semble réaliste, dans la mesure où un objet en mouvement se distingue du fond par une luminosité différente et donc des contours spatiaux, au moins sur une partie de sa frontière.

On envisage 2 phases d'ajustement :

- le mouvement du *snake* est contraint à faire partie de la hiérarchie de modèles de mouvement ;
- le mouvement est non contraint.

La première phase d'ajustement permet d'assurer une certaine cohérence entre le mouvement de la région elle-même et le mouvement de sa frontière. Notre implémentation s'est limitée à une contrainte de type affine. La deuxième étape est effectuée afin de faire adhérer le *snake* exactement au contour de la région et ne devrait fournir qu'un ajustement assez minime. Malheureusement, cette deuxième phase n'a pas été testée car notre implémentation ne prend en compte que des *snakes* à extrémités libres. Or nos régions ont absolument besoin de frontières fermées et la deuxième phase d'ajustement a tendance à séparer les 2 extrémités (ce qui n'est pas le cas avec la première phase, grâce à la contrainte affine).

3.3.1 Formulation mathématique du problème

Soit une courbe paramétrique plane fermée $S_0(s) = \begin{pmatrix} x_{S_0}(s) \\ y_{S_0}(s) \end{pmatrix}$; cherchons la transformation affine $\Theta^+ = [t_x^+, t_y^+, a^+, b^+, c^+, d^+]$ qui, appliquée à S_0 , minimise l'énergie suivante :

$$S(s) = \begin{pmatrix} x_S(s) \\ y_S(s) \end{pmatrix} = S_0(s) + \mathcal{M} \left[\Theta^+, \begin{pmatrix} x_{S_0}(s) - x_G \\ y_{S_0}(s) - y_G \end{pmatrix} \right]$$

$$E(S) = \frac{-1}{l_S} \oint_S \|\vec{\nabla} I[S(s)]\| ds$$

avec l_S la longueur du *snake*.

Remarquons qu'il n'y a pas de termes d'énergie interne au *snake* [TPBK87] [KWT88] [Ber91] :

- l'élasticité (coefficient α) est indésirable ici. Elle est classiquement utilisée pour réduire la longueur du *snake*, dans un contexte où il est initialisé manuellement à l'extérieur de l'objet dont on veut extraire le contour. Ici, l'initialisation pouvant se faire à l'intérieur de l'objet, il y a un risque de voir le *snake* se contracter jusqu'à devenir un point, car il n'est pas retenu dans son raccourcissement par le contour de l'objet.
- la rigidité (coefficient β) qui sert classiquement à assurer une certaine régularité à la courbe ne s'impose pas, car si S_0 est régulière S le sera aussi. De plus, si S_0 comporte un angle (discontinuité de la dérivée), cette discontinuité sera préservée.

3.3.2 Minimisation de l'énergie du *snake*

On se ramène donc au problème de minimisation non contrainte suivant :

$$\Theta_{\mathcal{R}, DSC}^+ = \arg \min_{\Theta^+} E(S)$$

qui est résolu par minimisation itérative à l'aide de la formule d'Euler-Lagrange³ :

$$\gamma_{\psi^+} \frac{d\psi^+}{d\tau} + \frac{\partial E}{\partial \psi^+} = 0$$

où

- ψ^+ désigne l'une des composantes de Θ^+
- τ désigne un "temps" qui repère l'évolution du *snake* dans l'image (temps local à I_{t+1} , qui n'a rien à voir avec le temps t de la séquence)
- γ_{ψ^+} est appelé "viscosité". En fait, ce paramètre s'interprète comme l'inverse du gain dans une méthode de minimisation par descente de gradient à gain constant :

$$\psi_{k+1}^+ - \psi_k^+ = \frac{d\psi^+}{d\tau} = \frac{-1}{\gamma_{\psi^+}} \frac{\partial E}{\partial \psi^+}$$

Cette équation se développe en

$$\frac{d\psi^+}{d\tau} = \frac{1}{l_S \gamma_{\psi^+}} \oint_S \left(\frac{\partial \|\vec{\nabla} I\|}{\partial x} [S(s)] \frac{\partial x_S(s)}{\partial \psi} + \frac{\partial \|\vec{\nabla} I\|}{\partial y} [S(s)] \frac{\partial y_S(s)}{\partial \psi} \right) ds$$

Les termes de dérivées spatiales de la norme du gradient s'interprètent comme des forces extérieures au *snake*, forces dérivant du potentiel $\|\vec{\nabla} I\|$ et se calculant directement dans l'image. Les autres dérivées partielles par rapport à ψ se calculent symboliquement à partir des équations du modèle de mouvement.

3.4 Estimation des paramètres du mouvement

À l'issue de l'ajustement de la segmentation spatio-temporelle, nous disposons des frontières des régions dans I_{t+1} ; il ne nous reste donc plus qu'à mesurer leurs mouvements pour alimenter le filtre de Kalman.

Cette mesure des paramètres du mouvement d'une région \mathcal{R} se fait par minimisation de l'EQM de reconstruction de cette région définie ainsi :

$$EQM_{\mathcal{R}_{t+1}} = \frac{1}{\#\mathcal{R}_{t+1}} \sum_{p \in \mathcal{R}_{t+1}} \left[\hat{I}_{t+1}(p) - I_{t+1}(p) \right]^2$$

avec \hat{I}_{t+1} l'image I_{t+1} reconstruite.

3. Cette dénomination vient d'une interprétation de ce problème en termes de mécanique des fluides : le *snake* est soumis à des forces extérieures (éventuellement à des forces intérieures) et se déplace dans un fluide visqueux.

$$\begin{aligned}
EQM_{\mathcal{R}_{t+1}} &= \frac{1}{\#\mathcal{R}_{t+1}} \sum_{p \in \mathcal{R}_{t+1}} \left[\hat{I}_t \left(p + \vec{d}_{t+1}^-(p) \right) - I_{t+1}(p) \right]^2 \\
&= \frac{1}{\#\mathcal{R}_{t+1}} \sum_{p \in \mathcal{R}_{t+1}} \left[\hat{I}_t \left(p + \mathcal{M} \left(\Theta_{\mathcal{R},t+1}^-, p \right) \right) - I_{t+1}(p) \right]^2
\end{aligned}$$

qui ne dépend que de $\Theta_{\mathcal{R},t+1}^-$. La minimisation se ramène alors à

$$\Theta_{\mathcal{R},t+1}^- = \arg \min_{\Theta_{\mathcal{R},t+1}^-} EQM_{\mathcal{R}_{t+1}}.$$

L'optimisation de $\Theta_{\mathcal{R},t+1}^-$ se fait par descente de gradient avec un gain adaptatif [Nic92]. L'inconvénient majeur de cet algorithme est que le résultat dépend fortement de l'initialisation à cause des minima locaux. Cette sensibilité aux conditions initiales justifie l'emploi d'une prédiction par filtre de Kalman.

3.5 Résolution des ambiguïtés

Jusqu'à présent, nous ne nous sommes intéressés qu'au suivi d'une seule région. Cette partie va traiter des problèmes qui se posent lorsque plusieurs régions sont suivies simultanément et qu'elles se chevauchent.

Il y a ambiguïté quand la segmentation en régions, à l'issue de la phase d'ajustement, ne forme pas une partition de l'image I_{t+1} . Les pixels ambigus se répartissent en 2 ensembles :

- les pixels qui n'appartiennent à aucune région : ce sont des pixels qui viennent d'être découverts par le mouvement d'une région, sans qu'une région déjà existante se trouve "en-dessous" à cet endroit. Une nouvelle région les contenant est créée ; elle pourra éventuellement être rattachée à l'une des régions voisines à $t + 2$. En fait, dans cette étude, ce cas n'a pas été envisagé puisque l'on considèrerait qu'il y avait toujours une région qui constituait le fond de l'image.
- les pixels qui appartiennent à plusieurs régions : ce sont des pixels où 2 régions ou plus se chevauchent. Ils sont rattachés à la région où ils sont le mieux reconstruits, qui doit être la région la plus "au-dessus". En fait, nous avons simplifié le problème en introduisant une connaissance *a priori* de l'ordre de superposition des régions.

Mise à jour du point G Quand une région passe en partie sous une autre, le centre de gravité de ses pixels visibles se déplace. Si l'on repérait le mouvement par rapport à ce point, qui est fortement modifié en cas d'occlusion, les paramètres de translation seraient biaisés et cela aurait des conséquences néfastes sur leur filtrage de Kalman. Pour contourner cette difficulté, le point G_t est mis à jour de la façon suivante :

$$G_{t+1} = G_t + \begin{pmatrix} t_{x,t}^+ \\ t_{y,t}^+ \end{pmatrix}$$

Ainsi, tant qu'il n'y a pas d'occlusion, le point G est confondu avec le centre de gravité (si le mouvement est bien estimé). En cas d'occlusion, il continue à se trouver là où il aurait été si l'occlusion n'avait pas eu lieu.

4 Résultats expérimentaux et extensions de l'algorithme

4.1 Présentation des séquences de test

Nous avons essentiellement testé l'algorithme sur des séquences d'images synthétiques. Celles-ci sont toutefois suffisamment critiques (amplitude importante des mouvements, occlusions fortes, suivi multiple) pour approcher des situations présentes en scènes réelles. Cela a plusieurs avantages :

- Chaque image se compose de régions bien identifiées dont la segmentation est connue. Une segmentation parfaite peut donc servir d'initialisation. Le nombre de régions peut être choisi faible, ce qui accélère les calculs.
- Le mouvement de chaque région est connu. Les paramètres estimés peuvent donc être comparés à chaque instant avec les paramètres réels. L'algorithme peut aussi être initialisé avec une valeur exacte pour l'aider à amorcer le filtre de Kalman.
- Le niveau de bruit dans chaque image peut être réglé.

Les figures 3 et 4 montrent la première et la dernière image des 2 séquences de test, en indiquant la numérotation des régions.

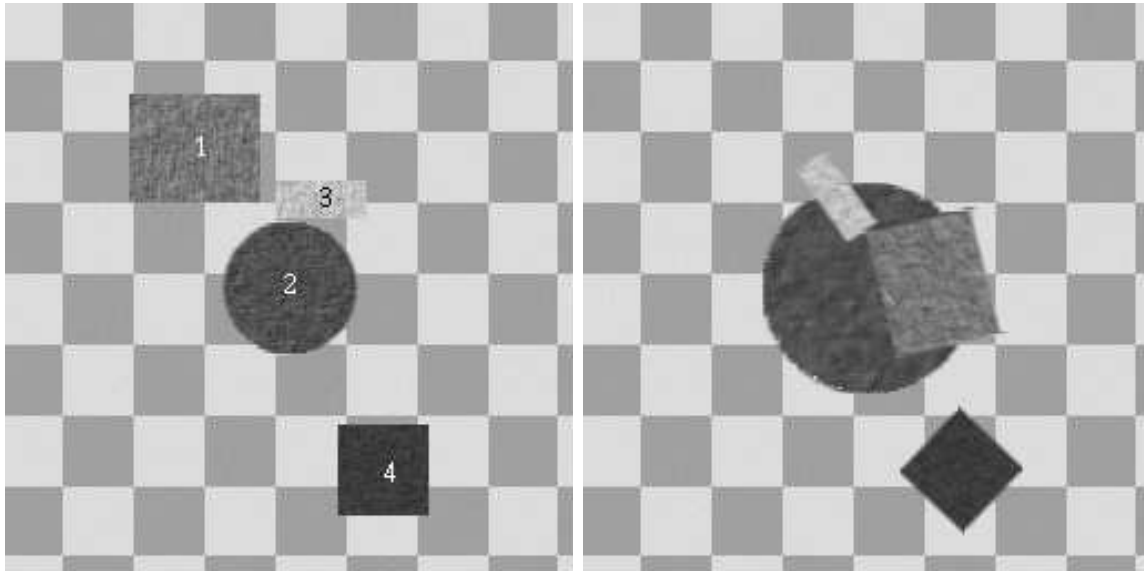


FIG. 3 - seqmill0 et seqmill19.

Séquence seqmill Il s'agit d'une séquence contenant des mouvements variés, simples ou composés, avec des régions qui rentrent en occlusion dès la 3^{ème} image.

INRIA

Elle se compose de 4 objets qui se détachent du fond par leur texture. Les paramètres de mouvement des régions de cette séquence sont les suivants :

numéro de région	t_x^+	t_y^+	k^+	θ^+
1	5	0	0	0.07
2	0	0	0.025	0
3	-2	0	0	0.05
4	0	0	0	0.04

Mais il faut noter que dans ces séquences, le point G qui sert de référence au mouvement est pris fixe, égal au centre de gravité de la région à $t = 0$. Donc, si un objet subit une rotation ou une dilatation non nulle, ses paramètres de translation ne restent pas constants au cours du temps.

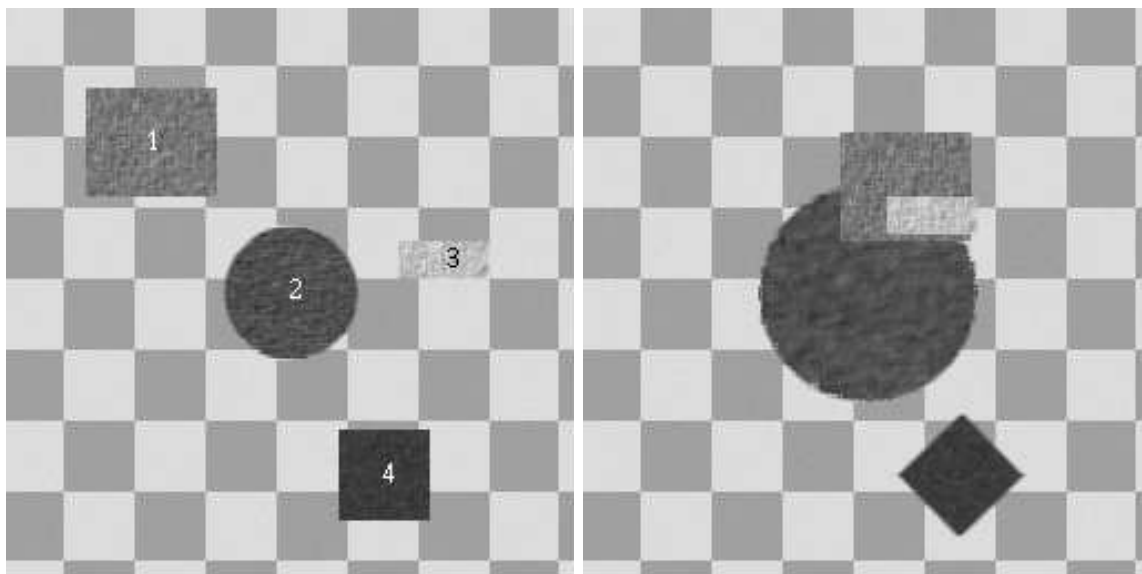


FIG. 4 - *seqmilt0 et seqmilt20.*

Séquence *seqmilt* Les objets de cette séquence sont les mêmes que dans *seqmill*. Seuls les positions initiales et les mouvements diffèrent. Cette séquence a été synthétisée pour tester l'algorithme dans des conditions plus favorables : les mouvements sont simples (translation, rotation ou dilatation pure) et un plus grand nombre d'images existent avant l'occlusion, afin de mieux amorcer le filtre de Kalman. Les paramètres de mouvement des régions de cette séquence sont les suivants :

numéro de région	t_x^+	t_y^+	k^+	θ^+
1	4	1	0	0
2	0	0	0.025	0
3	-2	-1	0	0
4	0	0	0	0.04

4.2 Snakes

Cette partie est consacrée à l'exposé de détails spécifiques à l'implémentation des *snakes* utilisée et à quelques résultats d'ajustement.

4.2.1 Implémentation

La courbe qui délimite chaque région est une B-spline de degré quelconque. Nous nous sommes limités au degré 1 (B-splines polygonales) car les objets ont des coins. Avec des B-splines de degré supérieur, il aurait fallu définir des points de contrôle multiples. Par ailleurs, l'initialisation peut se faire directement à partir de l'approximation polygonale fournie en entrée. D'autre part, des essais ont été faits pour approximer le polygone initial par une spline de degré quelconque, mais il apparaît des oscillations indésirables de la B-spline autour du polygone.

L'énergie du *snake* et la force qui lui est appliquée, sont calculées par discrétisation de chaque intervalle entre 2 points de contrôle en un nombre de points fixé à l'avance. La contribution de chaque point du *snake* discret est calculée par interpolation dans les images des forces ou de la norme du gradient.

Dans l'implémentation utilisée $\begin{pmatrix} x_G \\ y_G \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$; de plus, la minimisation de l'énergie du *snake* se fait séparément sur les paramètres (t_x, t_y) , en alternance avec les paramètres (a, b, c, d) . Ceci présente quelques inconvénients comme nous le verrons dans la partie sur les résultats globaux. Par exemple, le *snake* qui se trouve autour du carré en rotation est incapable de s'ajuster par simple rotation (voir les figures 17 et 18). Nous allons essayer d'expliquer ce phénomène par un modèle simplifié 1D à 2 paramètres k et t_x . Soit une image 1D dont la norme du gradient est représentée sur la figure 5.

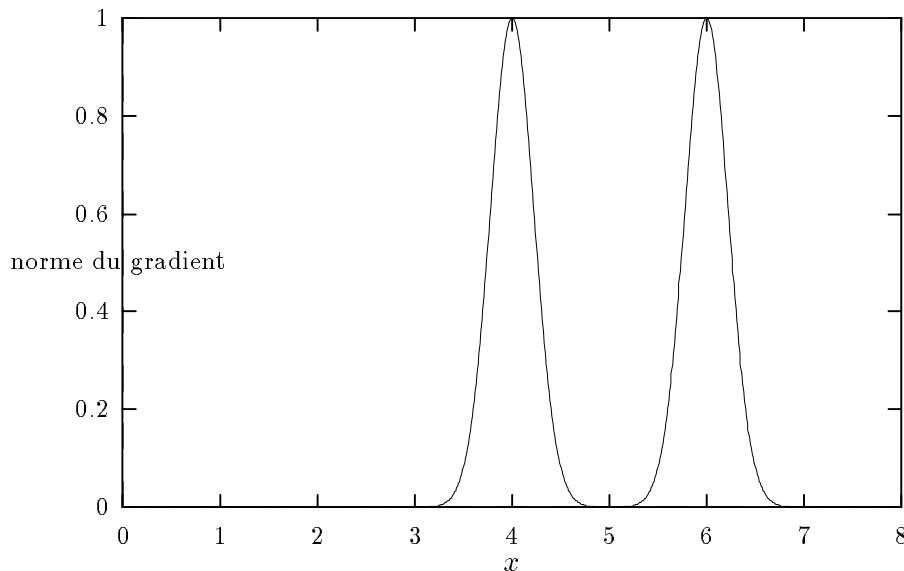


FIG. 5 - Norme du gradient dans une image 1D.

Initialisons un “*snake*” (en fait 2 points) en $x_1 = 4.1$ et $x_2 = 5.9$. Son centre de gravité se trouve en 5. Si $x_G = 5$, son déplacement affine idéal est une homothétie de

rapport $1+k = 2/1.8 \simeq 1.11$. Par contre, si $x_G = 0$ il faut composer la même homothétie par une translation $t_x \simeq -0.55$. Regardons maintenant les courbes d'énergie du *snake* en fonction des paramètres $(1+k, t_x)$ (voir les figures 6 et 7).

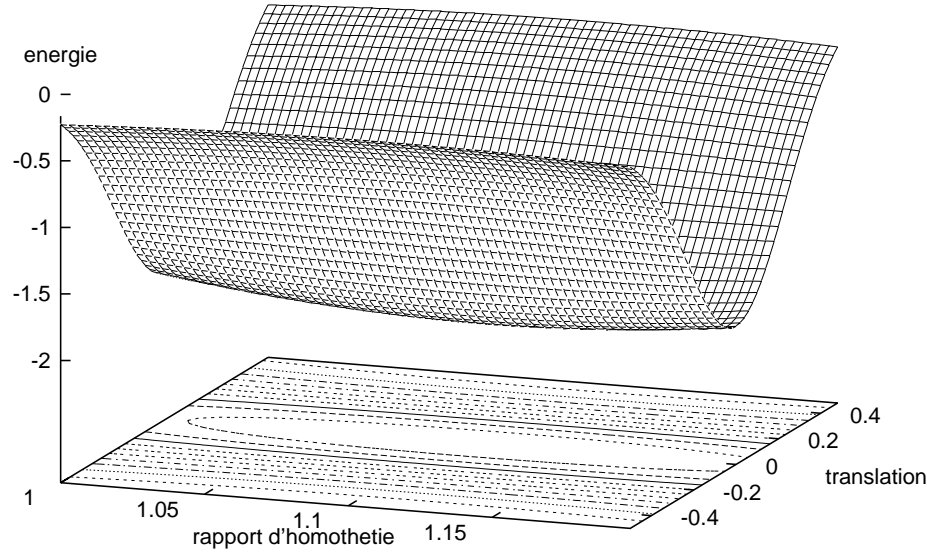


FIG. 6 - Énergie du *snake* avec $x_G = 5$: le minimum se trouve en $(1.11, 0)$.

La minimisation part du point $\begin{pmatrix} 1+k \\ t_x \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Dans le cas $x_G = 5$, la descente de gradient selon k nous amène au minimum global. Par contre, dans le cas $x_G = 0$, les minimisations successives selon t_x , puis selon k , ne nous font descendre que par très petits pas vers le minimum. Il faut un nombre d'itérations bien plus grand pour arriver au minimum global. En pratique, celui-ci n'est jamais atteint car il faudrait choisir un gain extrêmement faible, conduisant à un nombre d'itérations trop important (voir sur les figures 17 et 18 les cas où le *snake* entourant la région 4 a une rotation trop lente).

Pour résoudre ce problème, il faudrait donc

- soit choisir $\begin{pmatrix} x_G \\ y_G \end{pmatrix}$ proche du centre de gravité de l'objet,
- soit mener la minimisation simultanée sur les 6 paramètres affines.

4.2.2 Paramètre de lissage

La norme du gradient et ses dérivées spatiales sont calculées par un filtre optimal de détection de contours de type Canny-Deriche [Der87]. Un paramètre de ce filtre permet de choisir le degré de lissage du gradient. Il permet donc de régler la taille

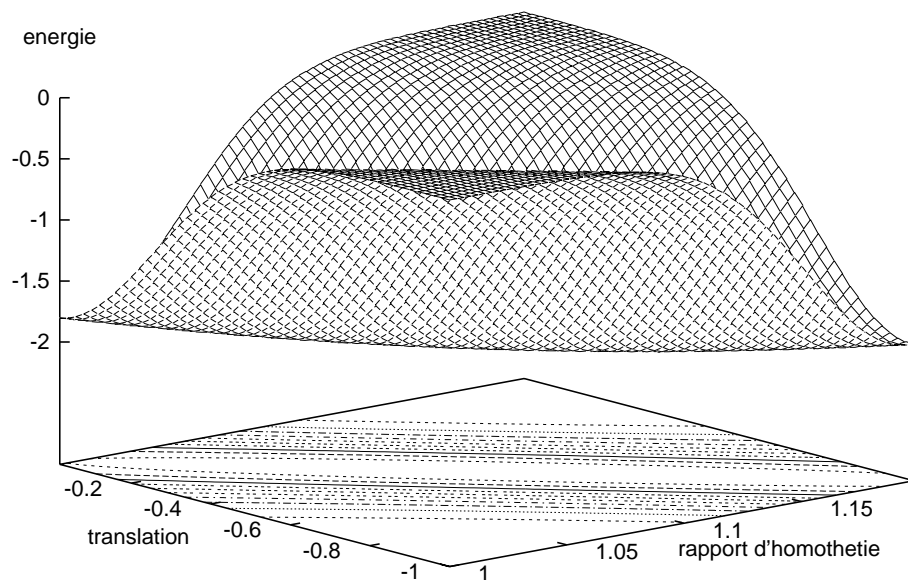


FIG. 7 - Énergie du snake avec $x_G = 0$: le minimum se trouve en $(1.11, -0.55)$.

de la zone où un contour est susceptible d'attirer le *snake*. Un lissage trop faible ne permet pas au *snake* de converger vers le contour s'il est initialisé trop loin. Un lissage trop important augmente le risque de voir le *snake* attiré par un contour voisin parasite (voir les figures 8 et 9).

Par exemple, sur la figure 8, le paramètre 0.7 provoque un lissage moyen. Le petit rectangle blanc est à peine discernable. Sur ces images, l'usage exclusif du gradient spatial constitue une perte d'information importante. En effet, la texture du rectangle qui lui permet de se distinguer du fond est considérée comme un bruit par le filtre de calcul du gradient.

Sur la figure 9, le paramètre 0.2 provoque un lissage très important. Le petit rectangle blanc est indiscernable.

4.2.3 Exemples d'ajustement

Les figures 10 et 11 montrent 2 exemples où l'ajustement est mal réalisé, à un instant donné, mais où l'erreur a été rattrapée 1 ou 2 images plus tard. Cela montre à la fois la sensibilité et la robustesse de l'ajustement par *snakes*.

Examinons en détail la figure 10. Le polygone entourant le rectangle a une petite irrégularité sur la gauche de son côté inférieur. Il s'agit de petits segments qui n'ont pas été éliminés par l'approximation polygonale initiale [IRE93a]. A cause du mode de calcul de la force (discrétisation de pas fixe), chacun de ces petits segments a autant de poids dans la force globale appliquée au *snake* que le grand segment qui délimite le côté supérieur du rectangle. Or ces petits segments sont attirés par le

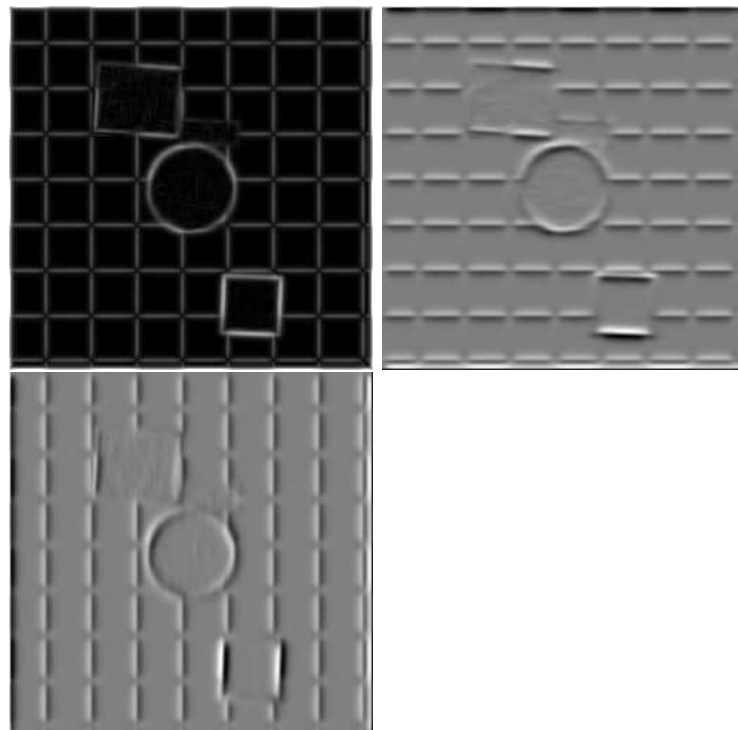


FIG. 8 - Image de la norme du gradient de `seqmill2` avec un paramètre de lissage de 0.7 et ses dérivées en x et en y (forces), lissées avec le même paramètre.

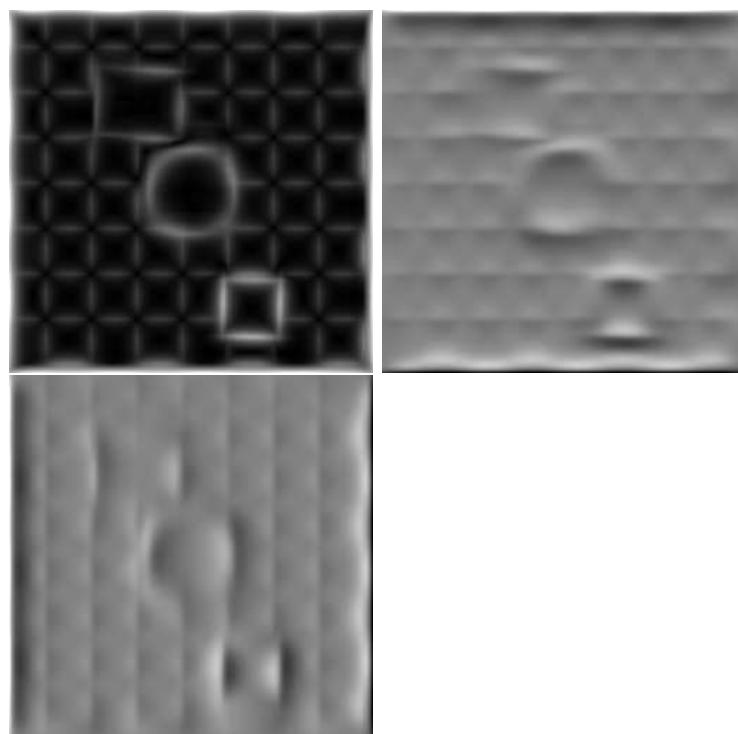


FIG. 9 - Image de la norme du gradient de `seqmill2` avec un paramètre de lissage de 0.2 et ses dérivées en x et en y (forces), lissées avec le même paramètre.

gradient qui est plus fort sur le damier du fond. Il est donc “normal” qu’ils attirent le *snake* dans son ensemble.

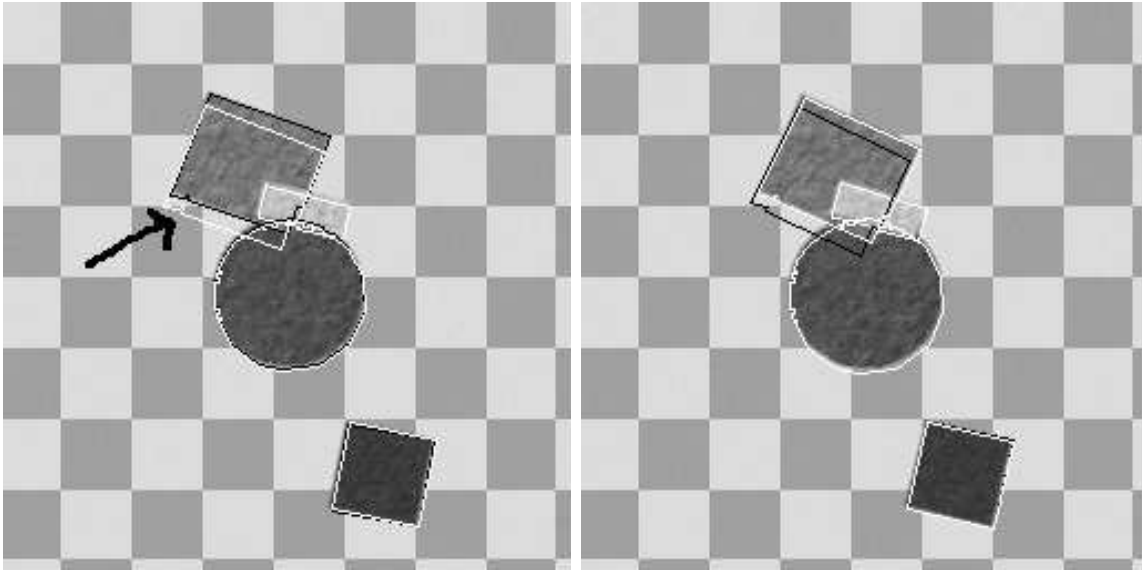


FIG. 10 - La segmentation prédite est en noir et la segmentation ajustée en blanc. Exemple de mauvais ajustement sur l’image `seqmilt5`, rattrapé dans l’image suivante. La flèche indique l’accumulation de petits segments sur la frontière du grand rectangle.

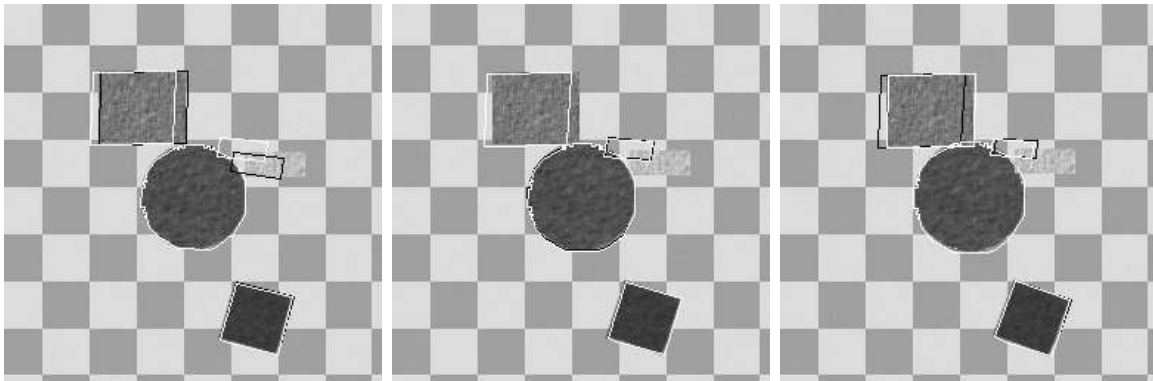


FIG. 11 - Autre exemple de mauvais ajustement sur l’image `seqmilt7`, qui se poursuit dans l’image suivante, mais qui est rattrapé dans `seqmilt9`.

Une amélioration suggérée par ce résultat est donc de choisir un pas de discrétisation adapté à la longueur de chaque segment. Ceci est aisé dans le cas d’une B-spline de degré 1, mais pour un degré plus élevé, la tâche est rendue plus délicate par le fait que la longueur de chaque intervalle du *snake* n’est pas connue à l’avance.

Par ailleurs, la région 3 est systématiquement mal ajustée. Son *snake* est attiré par les gradients spatiaux du fond qui sont plus forts que ceux de son contour. En conséquence, son EQM est très forte.

4.2.4 Conclusion

La difficulté principale concernant l'ajustement par contours actifs semble donc être leur comportement parfois chaotique, dû en partie à l'utilisation exclusive du gradient spatial. L'absence de filtrage temporel de l'ajustement⁴ fait qu'une erreur grossière n'est pas détectée tout de suite; il faut attendre l'image suivante pour qu'elle soit éventuellement rattrapée. L'amélioration du comportement des contours actifs pourrait donc consister en un couplage avec l'estimateur de mouvement :

- soit par une boucle ajustement \Rightarrow estimation itérée tant que l'EQM diminue,
- soit par une optimisation simultanée de l'énergie du *snake* et de l'EQM de la région. Pour cela, il faudrait reformuler l'énergie du *snake* en fonction de Θ_{t+1}^- au lieu de Θ_t^+ et minimiser une combinaison linéaire des 2 énergies.

L'avantage principal d'un ajustement contraint par rapport à un ajustement Markovien porte sur la quantité d'information à transmettre pour la segmentation. Dans notre cas, seul un vecteur de mouvement suffit. Par contre, dans l'autre cas, les frontières des régions subissent des variations aléatoires de quelques pixels [GL93]. Leur transmission nécessiterait alors un débit plus important, à moins de procéder à une régularisation temporelle.

4. Un tel filtrage est difficilement envisageable puisque par nature l'ajustement est imprévisible.

4.3 Filtrage de Kalman

4.3.1 Limites de la prédiction par filtre de Kalman

Divergence des paramètres de mouvement Les premiers essais qui ont été faits en suivant strictement le schéma de la figure 2 aboutissaient tous au même résultat : les paramètres de mouvement divergeaient. Cette divergence peut s'expliquer ainsi : la mesure de mouvement n'est pas indépendante de la prédiction par filtre de Kalman, parce que cette prédiction sert d'initialisation à la mesure et que l'estimateur de mouvement dépend fortement de la prédiction. Dès qu'une mesure s'écarte des précédentes, la prédiction pour l'instant suivant amplifie cet écart. S'il n'est pas dû à une vraie variation du mouvement de l'objet, mais à une variation aléatoire, alors la mesure se trouve mal initialisée. Dans ce cas, l'estimateur de mouvement tombe dans un minimum local très proche de la mauvaise initialisation. Ainsi, la prédiction se trouve auto-réalisée. Le phénomène peut donc se poursuivre avec une prédiction encore plus éloignée de la réalité : en quelques images, les paramètres de mouvement prennent des valeurs aberrantes.

Contrôle par initialisations multiples La solution adoptée consiste à ne pas faire confiance aveuglément à la prédiction. Plusieurs initialisations sont fournies à l'estimateur de mouvement. À chaque instant, nous disposons de 4 valeurs de paramètres de mouvement, de la plus simple à la plus élaborée :

- **initialisation par 0** : Dans le cas où la mesure précédente était complètement fausse, il peut être utile de revenir à cette valeur d'initialisation qui doit donner dans tous les cas une estimation raisonnable. Cela permet de rendre l'algorithme plus robuste en l'empêchant de diverger.
- **initialisation CTS (Court Terme Simple)** : On initialise avec la mesure précédente $\Theta_{t,EQM}^-$.
- **initialisation CTL (Court Terme Lissé)** : On initialise avec l'estimateur *a posteriori* précédent du filtre de Kalman, auquel est appliqué l'opérateur \mathcal{F} d'inversion des descripteurs de mouvement : $\mathcal{F}(\Theta_{t-1/t-1}^+)$.
- **initialisation LT (Long Terme)** : On initialise avec l'estimateur *a priori* courant du filtre de Kalman, auquel est appliqué l'opérateur \mathcal{F} : $\mathcal{F}(\Theta_{t/t-1}^+)$.

Chacune de ces initialisations nous fournit une EQM et une estimation. L'estimation qui donne la meilleure EQM est choisie comme mesure, ceci pour chaque région et à chaque image.

Voici des statistiques approximatives globales sur toutes les régions de toutes les images, recensant le nombre de fois où telle initialisation a été déclarée optimale :

meilleure initialisation	Séquence <code>seqmilt</code>	Séquence <code>seqmill</code>
LT	~ 40%	~ 30%
CTL	~ 30%	~ 30%
CTS	~ 20%	~ 30%
0	< 10%	< 10%

On voit qu'il n'y a pas de solution d'initialisation optimale; il semblerait donc qu'il faille conserver les 4 possibilités d'initialisation. Cette situation est plus onéreuse d'un point de vue calculatoire, même si les 4 voies peuvent être menées en parallèle. Mais on peut se demander si la différence d'EQM qui existe entre les 4 est significative; dans le cas contraire, il pourrait être envisagé de supprimer certaines possibilités en vue d'accélérer l'algorithme.

Pour cela, regardons la figure 12 montre les cas où le suivi s'est déroulé au mieux. L'initialisation par 0 est nettement moins bonne, ce qui démontre l'intérêt de faire un suivi temporel. Les 3 autres initialisations se comportent de façon presque identique. On pourrait donc éventuellement se contenter d'une initialisation (soit CTL, soit CTL) en plus de l'initialisation LT, tout en restant vigilant à la divergence des paramètres de mouvement.

Par contre, sur la figure 13, la prédiction long terme est mise en défaut pour l'image 12 et l'initialisation par 0 est la meilleure. Il s'agit d'un jeu de paramètres pour lesquels le suivi ne s'est pas déroulé au mieux. Dans ces situations critiques, il vaut mieux conserver les 4 possibilités.

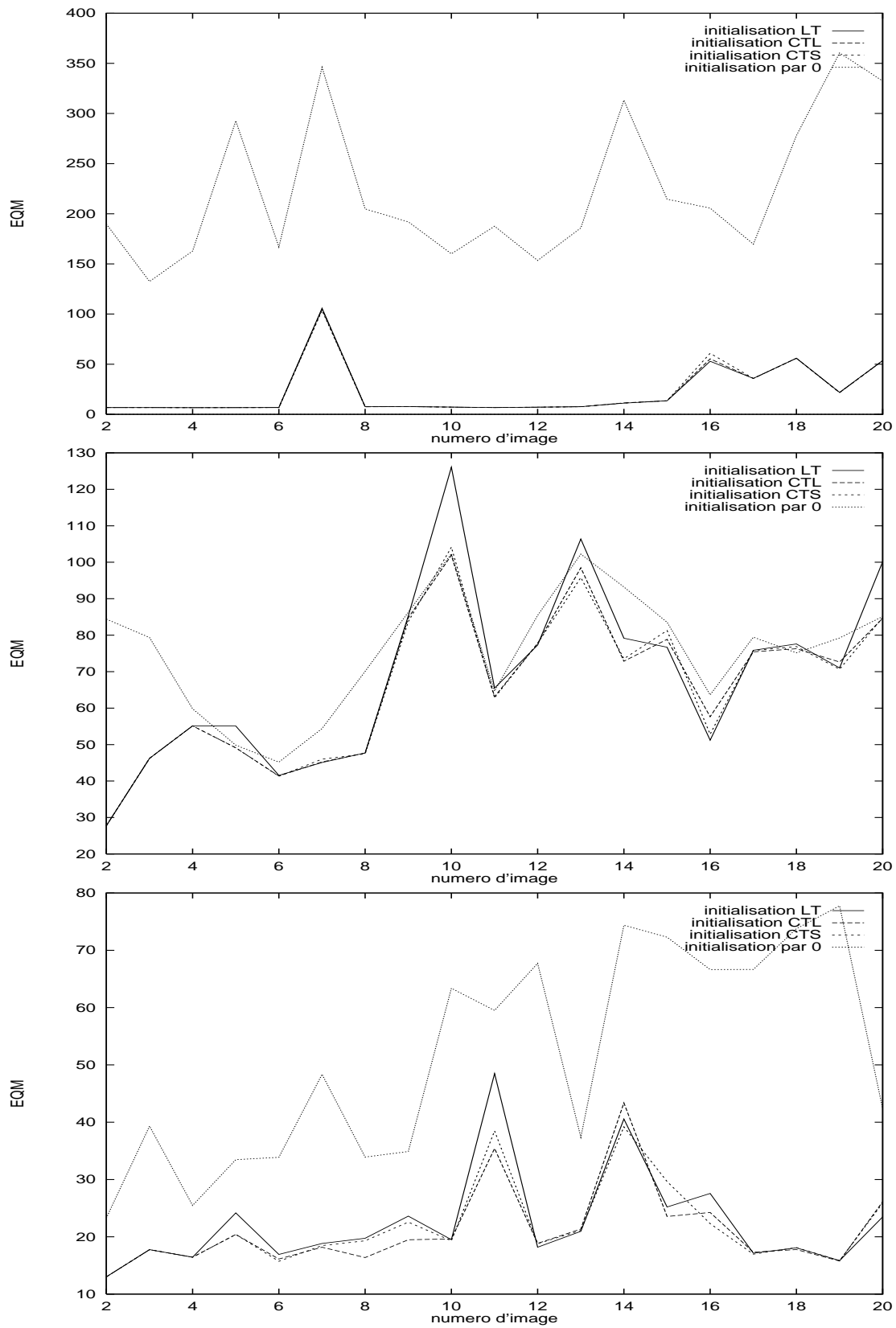


FIG. 12 - Exemples d'EQM pour les région 1, 2 et 4 de la séquence *seqmilt*. Il s'agit des résultats obtenus avec la valeur optimale des autres paramètres de l'algorithme.

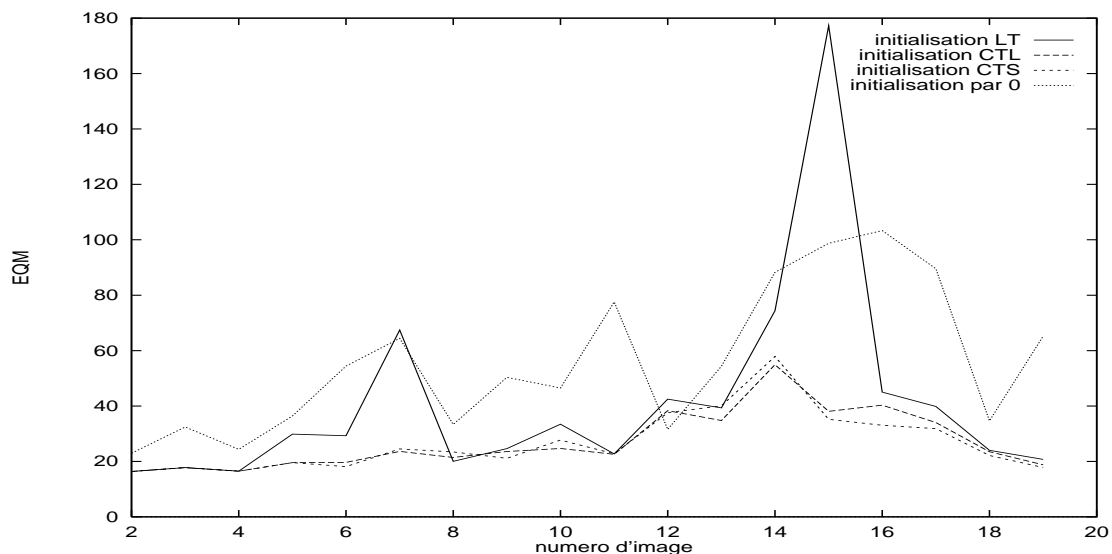


FIG. 13 - EQM pour la région 4 de la séquence `seqmill`. Il s'agit des résultats obtenus avec un paramètre de lissage de 0.9 et une "projection" CTS.

4.3.2 Courbes de filtrage

Dans cette section, nous allons regarder l'influence du filtrage de Kalman sur les paramètres de mouvement.

La figure 14 démontre l'intérêt de la prédiction. Celle-ci est presque toujours très proche de la bonne valeur de t_y à l'instant suivant.

Sur la figure 15, le paramètre t_x est constant donc la prédiction n'apporte rien par rapport aux mesures ou aux paramètres filtrés. Cet exemple suggère qu'il n'est pas pénalisant d'avoir un modèle d'évolution du filtre trop riche. L'apport du filtrage se voit à partir de l'image 13 où les mesures commencent à osciller. Les paramètres filtrés tendent alors à amortir ces oscillations.

Les courbes montrant des résultats similaires pour toutes les régions avec le jeu de paramètre optimal se trouvent en annexe A.

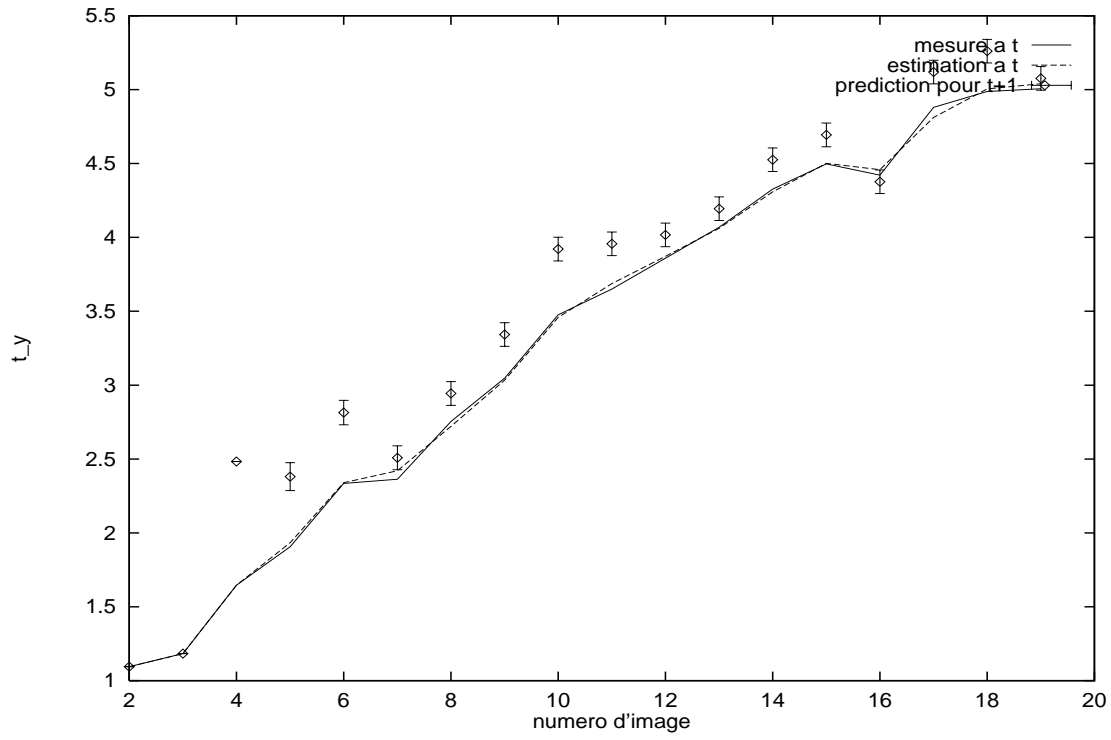


FIG. 14 - Paramètre t_y^+ de la région 1 de la séquence `seqmill`.

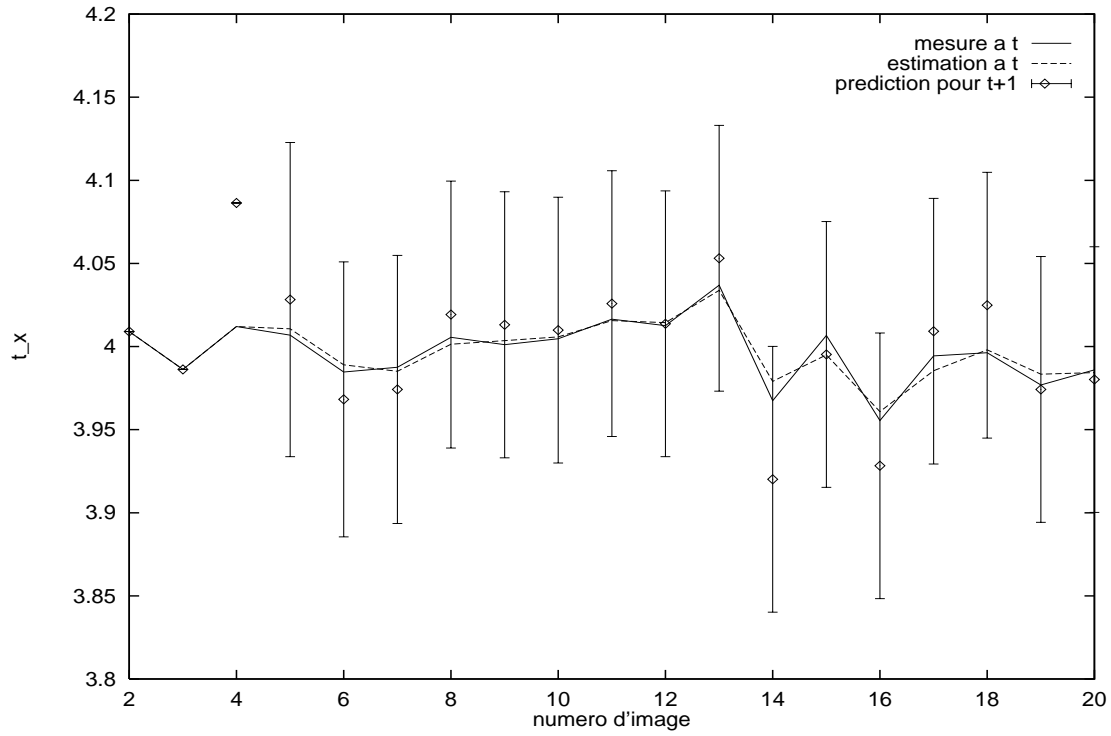


FIG. 15 - Paramètre t_x^+ de la région 2 de la séquence `seqmilt`. La valeur théorique est 4.

4.3.3 Comparaison de plusieurs politiques de “projection”

De même que l'on peut choisir plusieurs initialisations pour l'estimateur de mouvement, il est possible de choisir entre plusieurs prédictions pour la “projection”. Nous nous sommes limités aux 3 possibilités suivantes : CTS, CTL et LT. La projection par 0 n'a pas semblé intéressante ici car les *snakes* sont moins sensibles à une mauvaise initialisation que l'estimateur.

Mais, contrairement à ce qui se passe avec l'estimateur, il n'est pas si simple de choisir la meilleure “projection” *a posteriori*. D'une part, si l'on veut choisir la meilleure “projection” pour chaque région avant la résolution des ambiguïtés, le choix sera biaisé par les recouvrements entre régions. D'autre part, si l'on veut faire ce choix après la résolution des ambiguïtés, cette étape est rendue impraticable à cause d'une combinatoire explosive. Nous avons donc été contraints de choisir la politique de projection une fois pour toutes au début du programme.

La figure 16 et toutes celles de l'annexe B montrent les EQM obtenues pour chaque région, avec le jeu de paramètres optimal. Ces résultats sont à mettre en correspondance avec ceux de la partie 4.4 (voir les figures 17 et 18), où l'on montre les segmentations finales correspondantes. Le “pic” observé pour l'image 7 s'explique par un mauvais ajustement du type de celui montré sur la figure 10.

Là encore, telle ou telle projection n'est pas toujours la meilleure. Il semblerait donc que chacune ait son intérêt et que toutes méritent d'être conservées.

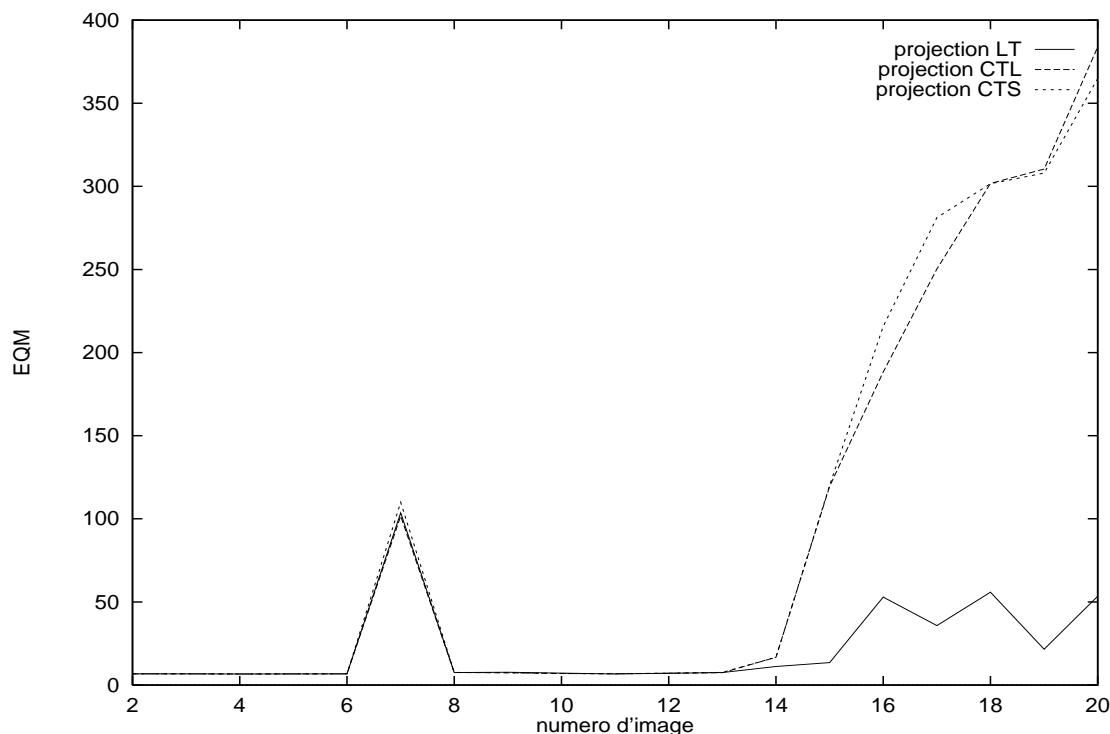


FIG. 16 - EQM pour la région 1 de la séquence seqmilt. Lissage de 0.5.

4.4 Résultats globaux

4.4.1 Meilleure EQM moyenne région par région

Les tableaux suivants résument sous forme numérique l'EQM de chaque région, moyennée sur la séquence entière, avec le jeu de paramètre optimal. Pour chacune, est indiqué le type de projection et le paramètre de lissage qui ont mené à ce résultat.

Les EQM sont bien meilleures pour la séquence `seqmilt` du fait de ses caractéristiques : mouvements plus simples et occlusions plus tardives. Les EQM obtenues pour la séquence `seqmill` laissent par contre encore à désirer. Notons toutefois que le mauvais suivi de la région 3 perturbe les estimations de mouvement des régions adjacentes 1 et 2. Si ce problème était résolu par les méthodes indiquées précédemment (voir § 4.2.4), le suivi de ces régions en serait sûrement grandement amélioré.

Séquence `seqmill`

numéro de région	meilleure EQM moyenne	écart type	projection	lissage
1	125.77	61.81	LT	0.9
2	60.90	18.42	LT	0.5
3	116.12	71.62	CTS	0.5
4	22.96	8.54	CTL	0.7

Séquence `seqmilt`

numéro de région	meilleure EQM moyenne	écart type	projection	lissage
1	22.38	25.74	LT	0.5
2	64.81	19.01	LT	0.5
3	83.26	52.48	LT	0.9
4	20.12	6.20	CTS	0.7

4.4.2 Segmentation finale

Les figures 17 et 18 montrent les 9 segmentations de la dernière image des 2 séquences, obtenues avec les 3 choix de la politique de projection (CTS, CTL ou LT) et 3 paramètres de lissage différents (0.5, 0.7 ou 0.9).

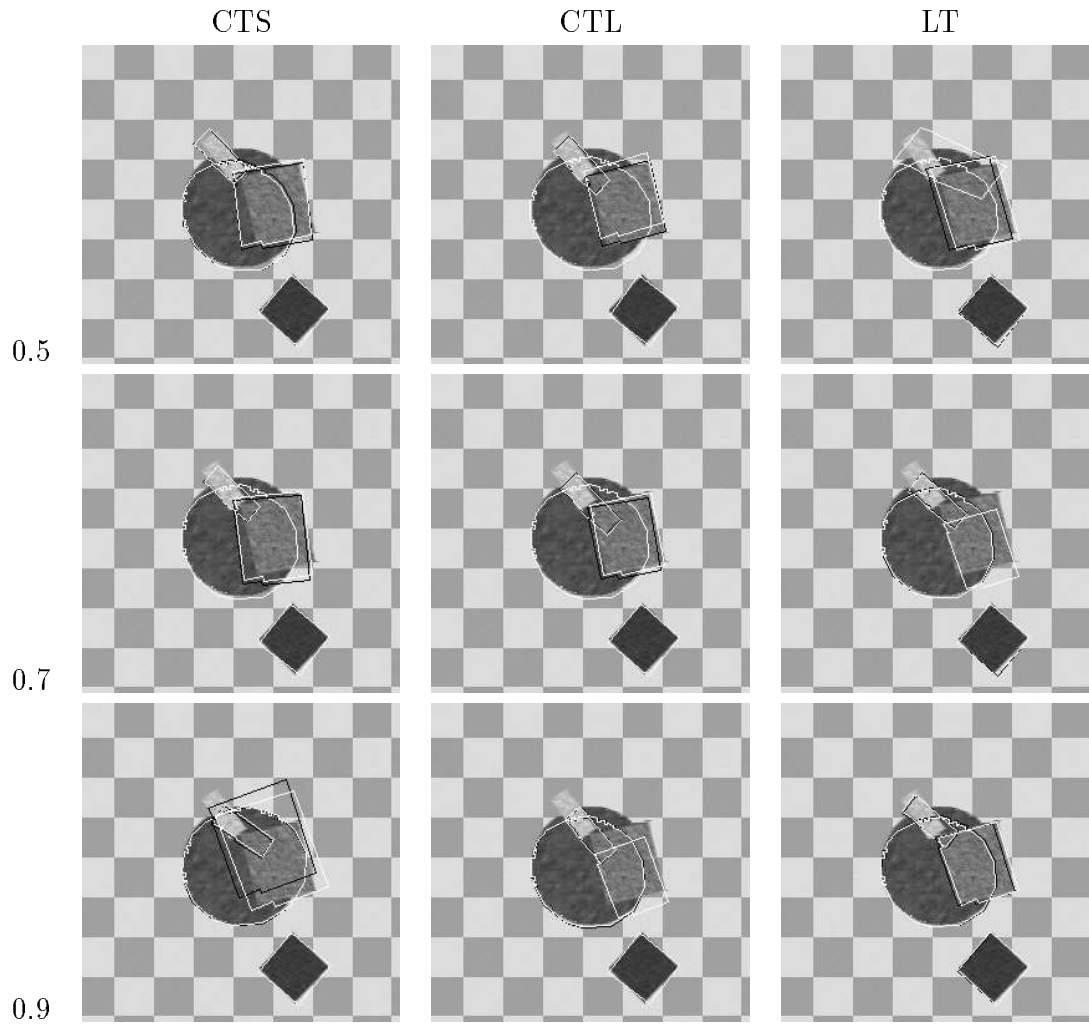


FIG. 17 - *Segmentation finale de seqmill19 selon le paramètre de lissage et la politique de projection choisis.*

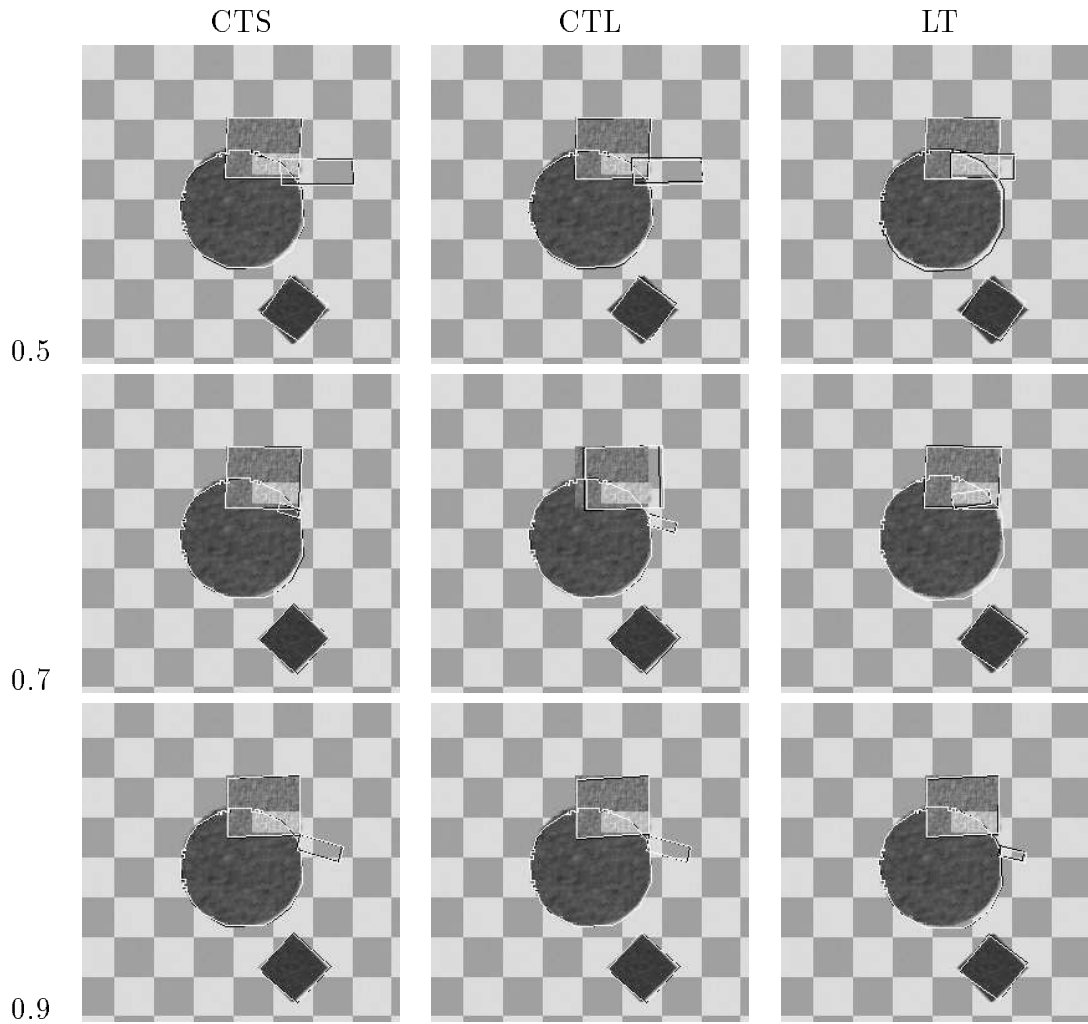


FIG. 18 - *Segmentation finale de `seqmilt20` selon le paramètre de lissage et la politique de projection choisis.*

5 Conclusion

Dans cette étude, nous avons donc conçu et développé un algorithme du suivi de segmentation en multiples régions homogènes au sens du mouvement, particulièrement bien adapté à la compression de séquences d'images numériques. Nous avons validé l'apport d'un suivi temporel et d'une prédiction par filtrage statistique sur des séquences synthétiques complexes, et obtenu des résultats préliminaires encourageants sur une séquence réelle. L'ajustement par *snakes* contraints présente un avantage potentiel du point de vue de l'encodage des frontières des régions ajustées, par rapport à un ajustement par champ de Markov. Nous introduisons comme élément de contrôle du suivi long terme un choix multiple d'initialisation de l'estimateur de mouvement.

Cependant, cette approche ne semble valable que pour un nombre restreint de régions suivies (scène de type visiophone par exemple). En effet, l'ajustement par contours actifs s'avère être efficace essentiellement lors du traitement de régions de taille significative. De plus, les EQM obtenues restent souvent fortes, notamment dans les zones d'occlusion où les frontières des régions sont mal ajustées et à cause du comportement parfois chaotique des *snakes*. On note par ailleurs une sensibilité de notre algorithme à la qualité de la segmentation initiale; cette remarque est malheureusement commune à toutes les méthodes de suivi.

Toutefois, la plupart de ces défauts pourrait être corrigée par les améliorations suivantes :

- choisir comme point de référence pour la déformation affine des snakes le même point que pour l'estimation de mouvement,
- prendre un pas d'échantillonnage adaptatif pour les B-splines,
- choisir *a posteriori* entre les 3 modes de "projection",
- coupler ajustement et estimation de mouvement, soit par une boucle ajustement \Rightarrow estimation, soit par une résolution simultanée des 2 problèmes.

D'autres extensions restent encore possibles :

- tester un modèle à vitesse constante pour le filtre de Kalman,
- prendre une hiérarchie complète de modèles de mouvement [Nic92], à la fois pour l'estimation de mouvement et pour l'ajustement,
- utiliser la segmentation de I_t dans l'estimation de mouvement pour ne pas perturber l'estimateur dans le cas d'une région découverte.

6 Annexe A : courbes des paramètres mesurés, estimés et prédits par filtrage de Kalman

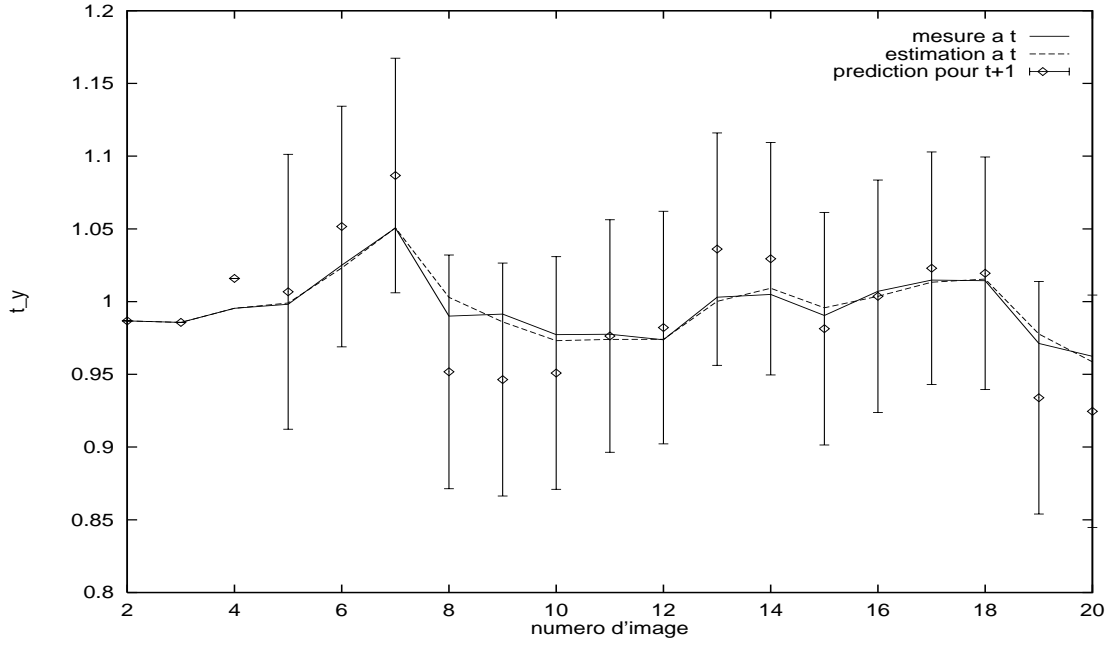


FIG. 19 - Paramètre t_y^+ filtré de la région 2 de la séquence `seqmilt`. La valeur théorique est 1.

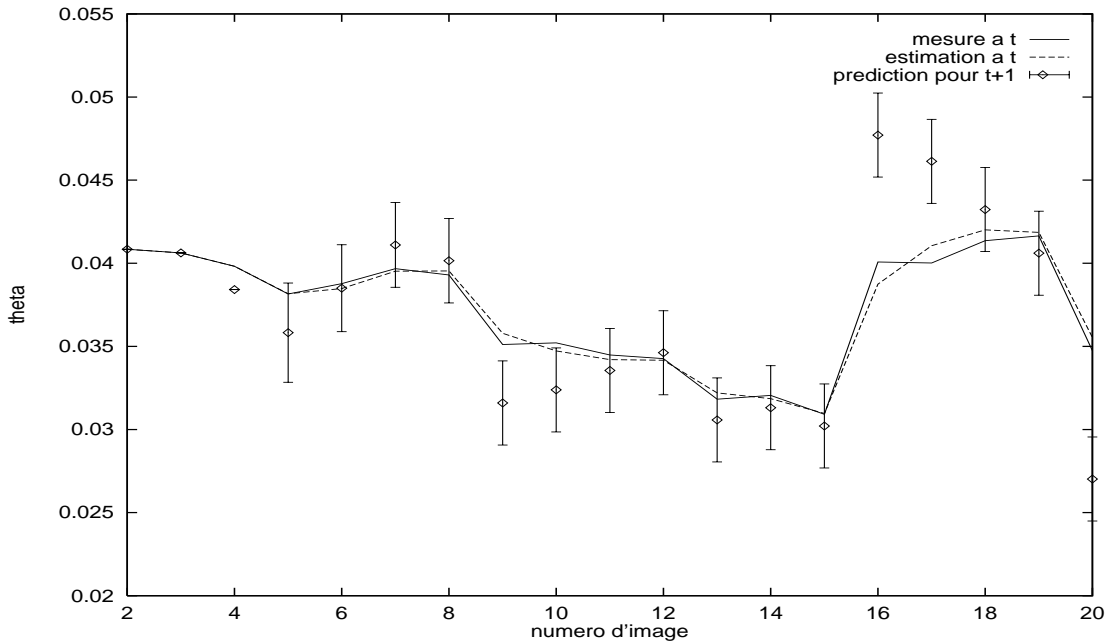


FIG. 20 - Paramètre θ^+ filtré de la région 4 de la séquence `seqmilt`. La valeur théorique est 0.04.

7 Annexe B : comparaison des “projections” CTS, CTL et LT

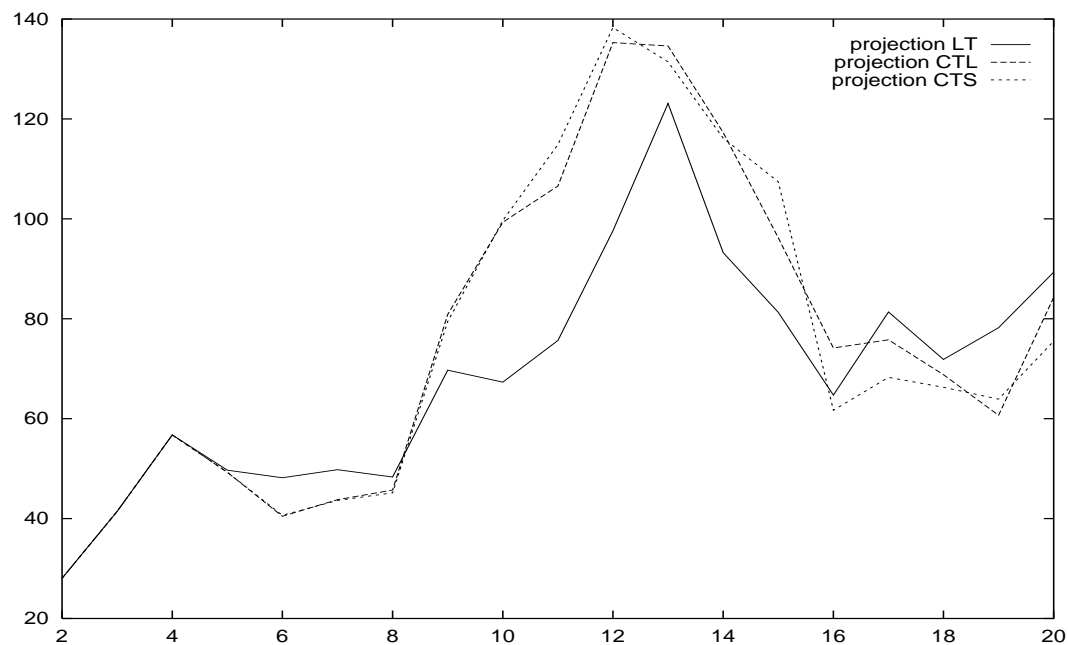


FIG. 21 - *EQM* pour la région 2 de la séquence `seqmilt`. Lissage de 0.7.

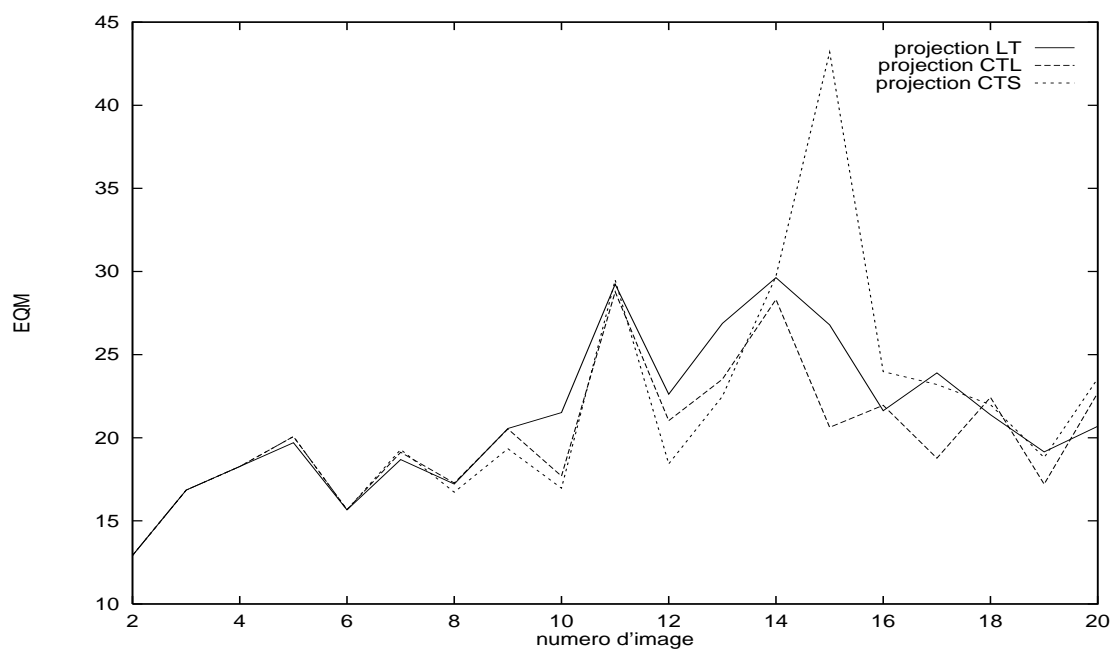


FIG. 22 - *EQM* pour la région 4 de la séquence `seqmilt`. Lissage de 0.5.

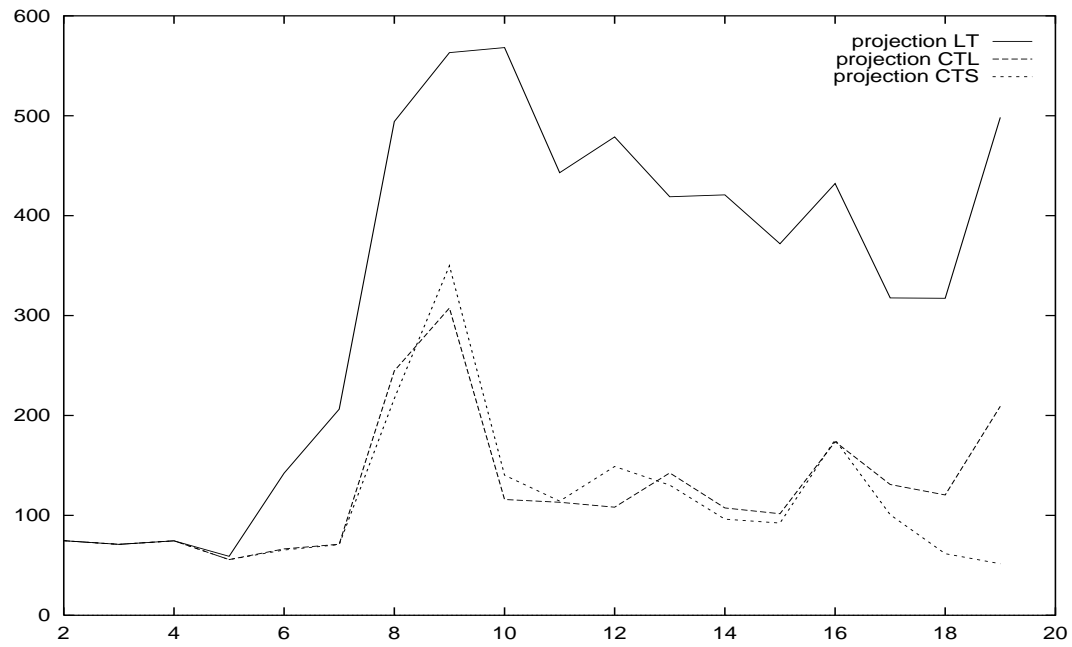


FIG. 23 - *EQM pour la région 1 de la séquence seqmill. Lissage de 0.5.*

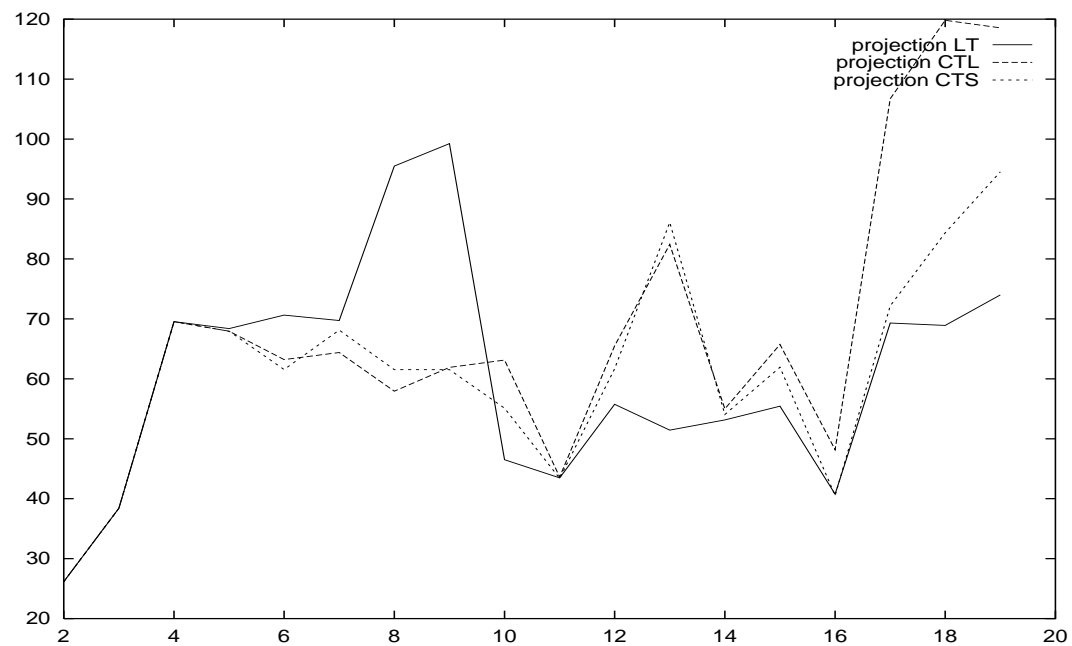


FIG. 24 - *EQM pour la région 2 de la séquence seqmill. Lissage de 0.5.*

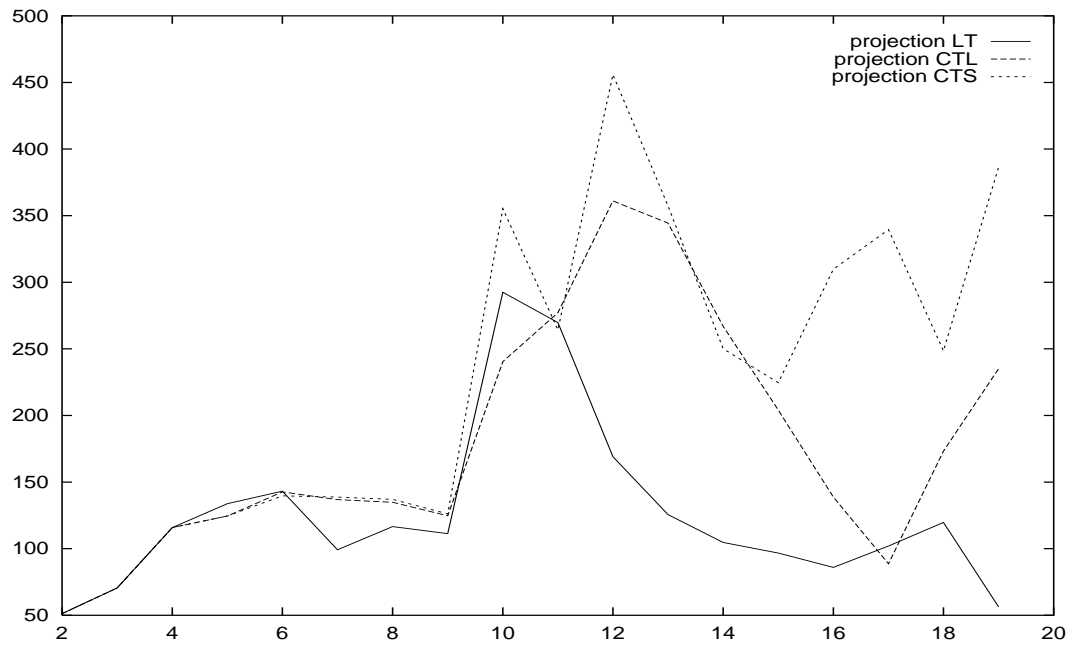


FIG. 25 - *EQM* pour la région 3 de la séquence **seqmill**. Lissage de 0.9.

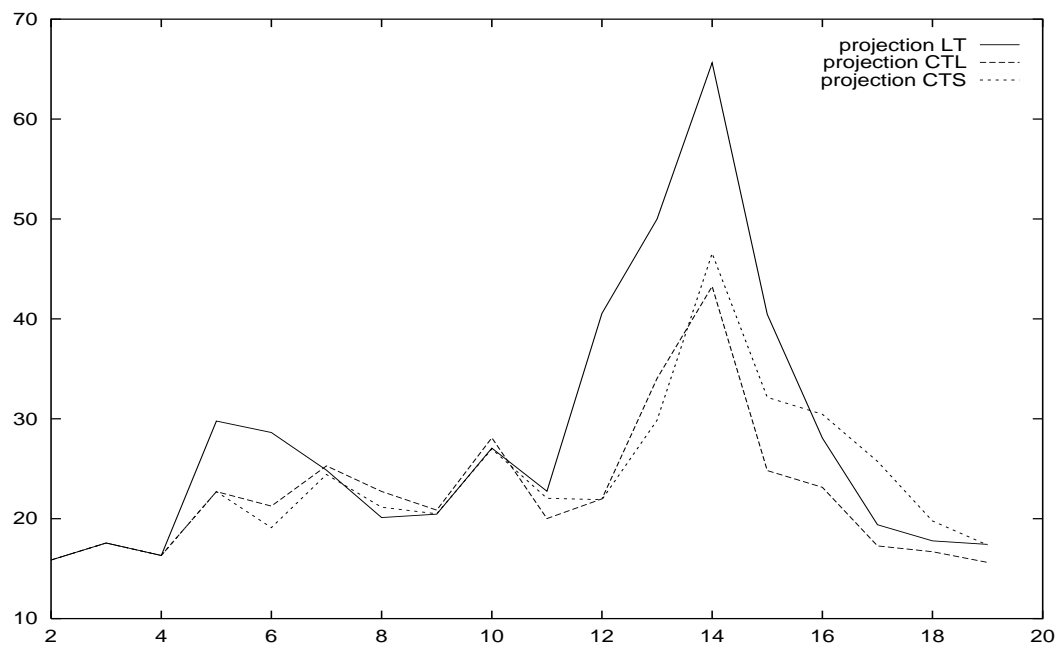


FIG. 26 - *EQM* pour la région 4 de la séquence **seqmill**. Lissage de 0.7.

Références

- [Adi85] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7(4):384–401, July 1985.
- [Ang92] P. Anglard. *Modélisation et contrôle des systèmes*. Cours de l'ECP, option mathématiques appliquées, Paris, 1992.
- [Bas92] Bénédicte Bascle. Motion-constrained tracking using deformable curves and applications. *Motion analysis and stereovision*, 1992.
- [Ber91] Marie-Odile Berger. *Les contours actifs : modélisation, comportements et convergence*. PhD thesis, INPL-INRIA Lorraine, February 1991.
- [BM93] Bénédicte Bascle and François Meyer. Suivi de primitives complexes dans une séquence d'images. In *ORASIS*, INRIA-Sophia, IRISA, February 1993.
- [Del93] Philippe Delagnes. *Contours actifs polygonaux*. Technical Report, IRESTE-LATI, June 1993.
- [Der87] Rachid Deriche. Using Canny's criteria to derive an optimal edge detector recursively implanted. *Int. Journal of Computer Vision*, 1(2):167–187, April 1987.
- [Fra91] Édouard François. *Interprétation qualitative du mouvement partir d'une séquence d'images*. PhD thesis, IRISA-Université de Rennes 1, Équipe TEMIS, June 1991.
- [Gal91] D. Le Gall. MPEG : a video compression standard for multimedia applications. *Communications of the ACM*, 34:46–58, April 1991.
- [Gal92] D. Le Gall. The MPEG video compression algorithm. *Signal processing : image communication*, 4:129–140, April 1992.
- [Gel74] A. Gelb. *Applied optimal estimation*. The MIT Press, Cambridge, 1974.
- [GL93] Victor Garcia-Garduño and Claude Labit. Suivi de segmentation spatio-temporelle pour la compression de séquences animées. In *Actes du GRETSI'93*, Juan les pins, September 1993.
- [IRE92] IRISA / IRESTE-LATI. *Rapport d'avancement numéro 1 - Segmentation spatio-temporelle pour le codage de séquences télévisuelles*. October 1992.
- [IRE93a] IRISA / IRESTE-LATI. *Rapport d'avancement numéro 2 - Segmentation spatio-temporelle pour le codage de séquences télévisuelles*. April 1993.
- [IRE93b] IRISA / IRESTE-LATI. *Rapport final - Segmentation spatio-temporelle pour le codage de séquences télévisuelles*. November 1993.

- [KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : active contour models. *International journal of computer vision*, 1:321–331, 1988.
- [LP80] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Roy. Soc. Lond.*, B-208:385–397, 1980.
- [Med69] J.S. Meditch. *Stochastic optimal linear estimation and control*. McGraw Hill, 1969.
- [Mey92] François Meyer. *Region-based tracking in an image sequence*. Technical Report, IRISA, July 1992.
- [Mey93] François Meyer. *Suivi de régions et analyse des trajectoires dans une séquence d'images*. PhD thesis, IRISA–Université de Rennes 1, Équipe TEMIS, June 1993.
- [Nic92] Henri Nicolas. *Hiérarchie de modèles de mouvement et méthodes d'estimation associées. Application au codage de séquences d'images*. PhD thesis, IRISA–Université de Rennes 1, Equipe TEMIS, September 1992.
- [NKL93] H. Nicolas, J. Konrad, and C. Labit. Joint estimation of motion and illumination variations for coding of image sequences. In *Proc. Scandinavian Conf. Image Analysis*, May 1993.
- [Ric92] Vincent Ricordel. Validation d'une hiérarchie de modèles de mouvements apparents 2d pour le codage de séquences d'images numériques. Rapport de stage - IRISA, 1992.
- [Sag89] C. Saguez. *Optimisation et contrôle optimal des systèmes*. Cours de l'ECP, option mathématiques appliquées, Paris, 1989.
- [San93a] H. Sanson. Estimation de mouvement basée sur des régions pour le codage de séquences télévisuelles. In *Actes du GRETSI'93*, Juan les pins, September 1993.
- [San93b] H. Sanson. Region-based motion estimation and compensation for digital tv sequence coding. In *PCS'93*, Lausanne, March 1993.
- [SC92] J. Shen and S. Castan. An optimal linear operator for step edge detection. *CVGIP: Graphical Models and Image Processing*, 54(2):13–17, March 1992.
- [Sin91] Ajit Singh. *Optic flow computation - A unified perspective*. IEEE Computer society press, 1991.
- [Sun93] Sun microsystems. The cell-a and cell-b image compression systems. 1993.
- [TPBK87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer graphics*, 21(4):205–214, July 1987.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)

ISSN 0249-6399